

Treball de Fi de Grau

## **Grau en Enginyeria en Tecnologies Industrials**

### **Millora i redisseny d'un robot Delta**

#### **MEMÒRIA**

**Autor:** Ferran Moyano i Tarrasón  
**Director:** Manel Velasco  
**Convocatòria:** Abril, 2018



Escola Tècnica Superior  
d'Enginyeria Industrial de Barcelona





# 1 Resum

Aquest treball consisteix en la millora en el funcionament d'un robot Delta amb l'objectiu d'aconseguir un prototip per a una aplicació del món industrial. Per aconseguir-ho s'ha treballat seguint els següents objectius:

- La comprensió de la mecànica del robot amb els seus components.
- La comprensió del funcionament dels diversos components electrònics.
- La realització d'un sistema de control i posicionament de l'efector final del robot.
- Aconseguir que el robot es mogui de forma suau i contínua.

L'aplicació industrial triada és la més habitual per aquest tipus de robot, la de "pick and place", que consisteix en agafar un objecte i deixar-lo en una posició desitjada. Als primers apartats es descriuen les diferents variants que es poden trobar en aquest tipus d'aplicació i s'ha limitat l'abast del projecte a l'aplicació "pick and place" per la qual l'objecte passa sempre per la mateixa posició amb la mateixa freqüència.

Per aconseguir els objectius s'ha estudiat la cinemàtica tant directa com inversa del robot. A partir de l'estudi de la cinemàtica s'han pogut optimitzar paràmetres importants per al nostre prototip, com ara el pla de treball, aquest s'ha determinat mitjançant un estudi de l'abast del robot.

Al determinar les posicions de treball del robot s'ha desenvolupat un codi que permet controlar el robot i enviar-lo a les posicions desitjades mitjançant unes trajectòries en línia recta, cosa que permet treballar a velocitat constant. Aquesta trajectòria es repeteix de forma repetitiva amb una freqüència que es pot fixar fàcilment per tal d'adaptar-la a la necessària per el sistema de treball.



# Índex

<b>1</b>	<b>Resum</b>	<b>1</b>
<b>2</b>	<b>Prefaci</b>	<b>6</b>
2.1	Origen del projecte . . . . .	6
2.2	Motivació . . . . .	7
2.3	Requeriments previs . . . . .	8
2.4	Objectius . . . . .	9
2.5	Abast del projecte . . . . .	10
<b>3</b>	<b>Introducció</b>	<b>11</b>
3.1	Tipus de robots . . . . .	11
3.2	Història del robot Delta . . . . .	12
3.3	Aplicació . . . . .	13
3.4	Normativa . . . . .	15
<b>4</b>	<b>Robot Delta</b>	<b>16</b>
4.1	Parts del robot Delta . . . . .	16
4.2	Cinemàtica i abast . . . . .	17
4.3	Cinemàtica inversa . . . . .	18
<b>5</b>	<b>El robot facilitat</b>	<b>20</b>
5.1	Base fixa . . . . .	21
5.2	Base mòbil . . . . .	22
5.3	Braç . . . . .	23
5.4	Avantbraç . . . . .	24
5.5	Actuador . . . . .	25
5.6	Motors . . . . .	26
5.7	Placa arduino . . . . .	28
5.8	Microstep drivers . . . . .	29
5.9	Font de voltatge . . . . .	31
5.10	Suport del robot . . . . .	32
<b>6</b>	<b>Disseny de l'actuador</b>	<b>33</b>
<b>7</b>	<b>Abast</b>	<b>35</b>
<b>8</b>	<b>Trajectòries</b>	<b>39</b>
8.1	Definició i paràmetres . . . . .	39
8.2	Algoritmes . . . . .	40
8.3	Trajectòria realitzada . . . . .	42
8.4	Altres possibilitats pel càlcul de trajectòries . . . . .	45
<b>9</b>	<b>Possibilitats de desenvolupament del treball</b>	<b>46</b>
<b>10</b>	<b>Conclusions</b>	<b>47</b>

<b>11 Annex</b>	<b>49</b>
11.1 Normativa . . . . .	49
11.2 Fitxes tècniques . . . . .	53
11.3 Codis . . . . .	64

## Índex de figures

1	Exemple d'un robot "pick and place" . . . . .	13
2	Exemples de diferents sistemes . . . . .	14
3	Esquema de les parts d'un robot Delta . . . . .	16
4	Canvi de Base . . . . .	18
5	Càlcul de l'angle . . . . .	19
6	Robot . . . . .	20
7	Base fixa . . . . .	21
8	Base movil . . . . .	22
9	Braç . . . . .	23
10	Avantbraç . . . . .	24
11	Taula d'especificacions dels motors . . . . .	26
12	Imatge obtinguda del llibre d'Arduino[1] . . . . .	27
13	Imatge obtinguda del llibre d'Arduino[1] . . . . .	27
14	Foto de la placa Arduino Uno . . . . .	28
15	2H microstep driver DQ420MA . . . . .	29
16	Microstep driver HY-DIV268N-5A . . . . .	30
17	Font de voltatge . . . . .	31
18	Especificacions . . . . .	31
19	Peça triada . . . . .	33
20	Actuador . . . . .	34
21	Vistes de la figura dissenyada . . . . .	34
22	Panoràmica: Punts als quals es pot situar l'efector del robot . . . . .	35
23	Alçat: Punts als quals es pot situar l'efector del robot . . . . .	36
24	Planta: Punts als quals es pot situar l'efector del robot . . . . .	36
25	Punts d'abast en el pla $z=-180\text{mm}$ . . . . .	37
26	Punts d'abast en el pla $z=-200\text{mm}$ . . . . .	37
27	Punts d'abast en el pla $z=-220\text{mm}$ . . . . .	37
28	Subjecció utilitzada . . . . .	42
29	Subjecció col·locada en el robot . . . . .	43
30	Taula pel càlcul de la trajectòria . . . . .	43

## 2 Prefaci

### 2.1 Origen del projecte

Aquest projecte parteix d'un altre treball realitzat per un estudiant de l'ETSEIB. El projecte previ consistia en el muntatge i el funcionament correcte del robot. El projecte que es presenta a continuació consisteix en l'optimització del funcionament del robot que s'havia fet prèviament.

Per tal d'optimitzar el robot cal analitzar característiques com l'abast i el rang de treball del robot per tal de triar sempre que sigui possible les condicions de treball més favorables. Les altres característiques que s'han d'optimitzar en un robot del tipus Delta tenen a veure amb el moviment d'aquest, cal triar la trajectòria adequada, la velocitat i l'acceleració.



## 2.2 Motivació

Personalment sempre he tingut interès en la programació ja que des de petit vaig aprendre a programar en diversos llenguatges per tal d'anar desenvolupant diferents tipus de funcionalitats, amb aquest projecte apareixia el repte d'aprendre la programació amb la IDE (integrated development enviroment") d'arduino. A més cal puntualitzar que l'ús que havia donat a la programació en general eren de caràcter pedagògic i al realitzar un treball d'aquest caire estaria focalitzat a l'aplicació en un automatisme real.

L'interès en la robòtica el vaig torbar a través de l'assignatura anomenada Projecte1, on vaig triar com a projecte la realització d'un robot. Després de realitzar l'assignatura no vaig seguir treballant en aquest camp tot i que l'interès seguia existint. Personalment crec que el món de la robòtica està creixent de forma exponencial i que s'està incloent en la vida quotidiana a grans passes, per tant tenir un coneixement de la realització i la programació en robòtica és de gran interès.

Actualment degut a les grans facilitats que es troben al mercat per a l'obtenció de materials i tecnologies cada vegada hi ha més gent que pot crear els seus propis robots o millorar-ne els obtinguts. Un exemple molt clar són les impressores 3D, aquestes impressores no deixen de ser un robot, és molt típic fer ús de la impressora obtinguda per tal d'imprimir noves peces que permetran que el funcionament sigui de millor qualitat o adaptar-la a noves facilitats.

A més cal destacar que actualment es troben multitud de plataformes del que es coneixen com a plataformes de codi obert per al desenvolupament de programes de control. Els costos d'obtenció de motors, controladors, plaques... s'han vist molt reduïts en els darrers anys, això és un dels principals motius per els quals la robòtica s'ha començat a veure com una curiositat personal i ja no es veu només com una activitat professional.

## 2.3 Requeriments previs

Per a poder realitzar el treball cal tot el material per a fer el robot o, com en el cas d'aquest treball, el robot ja fet. El material inclou tant el que cal per a obtenir la forma d'un robot, com els elements per a fer-lo funcionar, que seria una part més electrònica.

A part dels materials també és necessari tenir coneixements en programació per a poder realitzar tota la part del control del robot, en matemàtiques i física per poder realitzar l'estudi de la cinemàtica

## 2.4 Objectius

L'objectiu principal d'aquest treball és la possibilitat de controlar el moviment d'un robot tipus delta, per tal de que el seu efector final vagi a una posició desitjada i trobar una funcionalitat al robot.

Els objectius específics són:

- La comprensió de la mecànica del robot amb els seus components.
- La comprensió del funcionament dels diversos components electrònics.
- La realització d'un sistema de control i posicionament de l'efector final del robot.
- Aconseguir que el robot es mogui de forma suau i contínua.

## 2.5 Abast del projecte

El projecte consisteix en aconseguir que el robot Delta facilitat per el departament sigui capaç de realitzar la funcionalitat coneguda com a "pick and place", on el robot és capaç d'agafar un objecte i transportar-lo a un altre lloc. Com s'explicarà en futurs apartats aquesta funció pot tenir diversos graus de dificultat depenent d'on s'utilitzi el robot. L'objectiu principal és que el robot sigui capaç d'agafar una peça que sempre està a la mateixa posició i té la mateixa freqüència de pas. Un cop s'assoleixi aquest objectiu es podria anar ampliant el treball amb millores dintre de la funcionalitat.

El treball també té com a objectiu l'estudi dels possibles plans de treball del robot i la tria del pla més òptim de treball. L'estudi de trajectòries i el plantejament d'algoritmes per a solucionar la problemàtica explicant les seves limitacions.

## 3 Introducció

### 3.1 Tipus de robots

Els robots es poden classificar segons la seva estructura, podem trobar robots d'estructura en sèrie i d'estructura paral·lela. Els robots d'estructura en sèrie són els més antics i es defineixen per tenir una cadena cinemàtica oberta, en canvi els robots d'estructura en paral·lel tenen dues bases, una de fixa i una de mòbil que estan unides per diverses cadenes cinemàtiques en paral·lel que en el seu conjunt formen una cadena cinemàtica tancada. El robot Delta forma part d'aquest segon tipus de robots.

Les característiques del robot en paral·lel en comparació en un robot en sèrie són les següents. El principal inconvenient és que tenen un espai de treball menor. L'avantatge principal és que la relació de la massa del robot en front a la de la càrrega és menor, d'aquesta característica sorgeixen les propietats més interessants dels robots en paral·lel, aquestes propietats són majors acceleracions i velocitats en el moviment i una major precisió. Un altre avantatge és la col·locació dels motors que en un robot paral·lel són fixos en la plataforma fixa, mentre que en un robot tipus sèrie té els motors col·locats a les articulacions.[2]

### 3.2 Història del robot Delta

El robot Delta va ser dissenyat per Reymond Clavel al començament dels anys 80 a la universitat de Lausanne a Suïssa. Com s'extreu de la seva tesi "Conception d'un robot parallele rapide à 4 degrés de liberté" publicada a l'any 1991, el robot va sorgir de la motivació de solucionar una necessitat industrial d'aquella època, el tractament de peces lleugeres (uns pocs grams) a una cadència elevada.

El projecte del robot s'origina quan una empresa de bombons visita a R.Clavel, l'empresa va demanar ajuda per intentar solucionar la problemàtica generada per col·locar els bombons a les caixes. Es van trobar dos problemes, el primer era que l'activitat era monòtona i difícil per els operaris, el segon era que aquesta manipulació dels bombons per part dels operaris era vista com a poc higiènica. Com a objectiu per a resoldre aquesta problemàtica es va fixar que calia que el robot fos capaç de treballar a una cadència superior a la dels operaris.[3]

El funcionament i la geometria dels robots Delta actuals i el proposat per R.Clavel no tenen gaires diferències. R.Clavel va dissenyar un robot Delta amb quatre graus de llibertat, els tres de translació més un de gir en l'eix vertical, actualment es poden trobar robots Delta amb només 3 graus de llibertat que seran sempre els corresponents a les translacions. El nombre de graus de llibertat vindran definits per l'ús que se li donarà al robot. La majoria de canvis venen deguts a la possibilitat de treballar amb motors més potents i més precisos i a la utilització de materials més lleugers.

### 3.3 Aplicació

L'aplicació triada és coneguda com a "pick and place", aquesta tasca es troba usualment en fàbriques que han d'empaquetar els seus productes per tal de distribuir-los en lots. La tasca "pick and place" consisteix en agafar objectes d'una cinta transportadora i col·locar-los en una posició desitjada d'una capsa o paquet. Un exemple pot ser el d'una fàbrica de xocolatines on caldrà agafar cadascuna d'aquestes de la cinta transportador i col·locar-les en un paquet.

El o els robots que s'utilitzin hauran d'adaptar-se al sistema "pick and place" dissenyat.



Figura 1: Exemple d'un robot "pick and place"

El sistema normalment consta d'una cinta transportadora principal de la qual els objectes són agafats i d'una secundària on es col·loquen. El sistema haurà de poder controlar i coordinar el flux d'objectes que van per la cinta transportadora principal i el flux de paquets per la segona cinta transportadora. Per exemple si un paquet té capacitat per a 6 objectes caldrà que quan s'hagin agafat i col·locat 6 objectes la cinta secundària avanci l'equivalent a un paquet.

El sistema pot tenir diversos graus de dificultat, que dependran principalment del nombre de robots, del flux dels objectes i de la col·locació d'aquests. El sistema augmentarà en complexitat si el nombre de robots és major ja que caldrà coordinar-los per tal de saber quants objectes han sigut col·locats.

En quant al flux d'objectes per la cinta principal podem trobar diverses configuracions possibles, la més senzilla seria on la freqüència de pas dels objectes és constant i la posició de la qual s'agafaran també com en la figura 2a, les altres configuracions

consistiran en variar la freqüència com en la figura 2b i variar la freqüència i la posició com en la 2c.

La col·locació dels objectes en la o les cintes secundàries també influirà en la dificultat del sistema. La dificultat del sistema de col·locació pot venir donada per diversos factors, un d'ells és el d'incloure més d'una cinta on segons el tipus d'objecte cal col·locar-lo en una cinta o una altra, en aquest cas el robot també faria una funció de selector. L'altre factor que influirà en el sistema de col·locació serà si l'objecte sempre és deixarà en una posició constant o si per altra banda caldrà que la posició variï.

En el conjunt el sistema haurà de tenir en compte totes les components mencionades anteriorment per a coordinar-les per tal d'aconseguir-ho, el disseny del sistema, en la majoria de les situacions, necessitarà de sensors per tal de poder situar els objectes en la cinta principal i d'un sistema de control per a poder saber la col·locació dels objectes. [4]

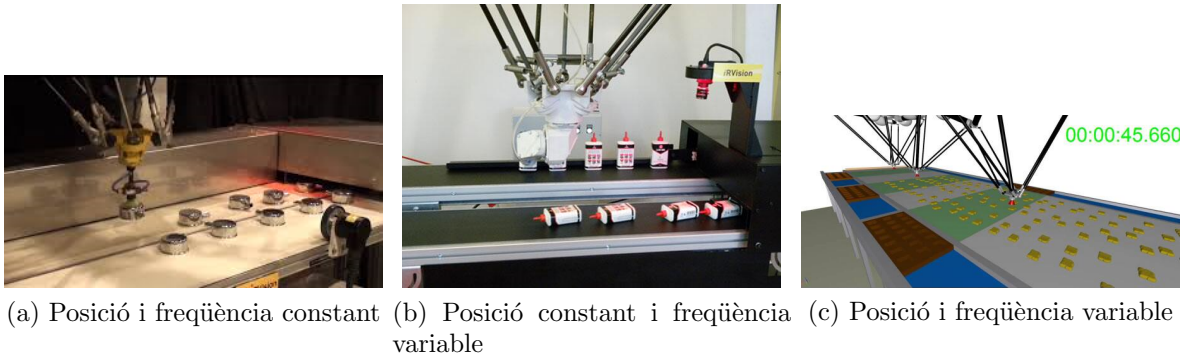


Figura 2: Exemples de diferents sistemes



### 3.4 Normativa

Al haver triat una aplicació de caire industrial, és interessant conèixer la normativa vigent. La normativa per a la seguretat dels robots industrials es regeix per la normativa ISO 10218-1:2011, el títol de la norma és el següent, “Robots i dispositius robòtics, Requeriments de seguretat per robots industrials” [5]. Per poder accedir al document s’ha de realitzar un pagament, però a través d’un document realitzat per l’AENOR (Asociación Española de Normalización y Certificación) que es troba a l’annex.

Aquesta norma descriu els riscos bàsics associats als robots i proporciona els requisits per a l’eliminació o la reducció adequada dels perills associats a aquests riscos. Si s’observa l’índex del document es poden identificar els paràmetres que requereixen d’un sistema de control per a poder utilitzar un robot en l’àmbit industrial. Els paràmetres són l’accionament del robot, el funcionament d’aturada del robot, la velocitat i els possibles moviments simultanis. A més s’apliquen restriccions en el funcionament del propi robot.

## 4 Robot Delta

### 4.1 Parts del robot Delta

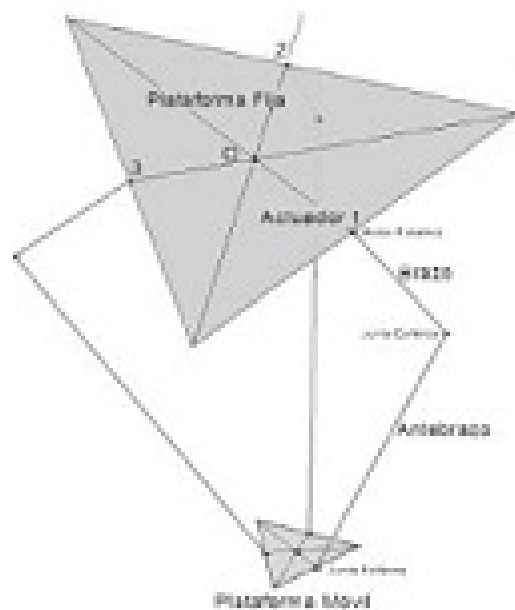


Figura 3: Esquema de les parts d'un robot Delta

El robot de tipus Delta com ja s'ha dit prèviament és un robot d'estructura paral·lela. Consta de dues bases, una de fixa on es troben els 3 motors que controlen la posició de l'extrem del robot i una de mòbil que és on hi haurà l'actuador en el cas de necessitar-ne un. Les bases acostumen a tenir una forma triangular tot i que no és absolutament necessari, la col·locació dels motors ha de fer-se de tal manera que formin un triangle equilàter entre ells, és per facilitar això que s'acostumen a utilitzar bases triangulars, de la mateixa manera cal que els punts d'unió entre la base mòbil i les barres també formin un triangle equilàter.

Les dues bases estan unides mitjançant 6 barres agrupades en dos grups, les que estan en connexió amb la base mòbil, anomenades avantbraç, i les que estan en connexió amb la base fixa, anomenades braç. Cada parella de barres estarà unida mitjançant una ròtula esfèrica. Aquests dos grups de barres poden, o no, tenir la mateixa mida.

Les dimensions de les barres i de les bases definiran l'abast que tindrà el robot. Les variables que quedaran definides al muntar el nostre robot seran: ' $f$ ': el costat de la base fixa, assumint que la base té la forma d'un triangle equilàter, ' $e$ ': el costat de la base mòbil, assumint també que la base mòbil té forma triangular, ' $rf$ ': la longitud dels braços i ' $re$ ': la longitud dels avantbraços.

## 4.2 Cinemàtica i abast

La cinemàtica d'un robot Delta es pot descriure de forma directa i de forma inversa. Per tal de definir la cinemàtica caldrà agafar un punt d'origen i caldrà definir uns eixos de coordenades cartesianes. La cinemàtica directa ens indica en quina posició  $(x,y,z)$  es trobarà l'efector final al girar els motors uns graus determinats  $(\theta_1, \theta_2, \theta_3)$ , si ens trobem en aquest cas les variables dependents són  $(x,y,z)$  i les independents són  $(\theta_1, \theta_2, \theta_3)$ . El cas més interessant és el de la cinemàtica inversa ja que indica els graus  $(\theta_1, \theta_2, \theta_3)$  que cal girar cadascun dels motors per tal d'aconseguir que l'efector final es trobi en una posició  $(x,y,z)$  desitjada, per tant en aquest cas les variables que abans eren dependents ara són les independents i viceversa.

Per tal de calcular l'abast del robot es realitzarà una simulació de les possibles posicions a les quals el robot pot accedir i s'obtiniran els gràfics corresponents. Aquesta simulació es pot fer mitjançant la cinemàtica directa i un codi informàtic. Conèixer l'abast del robot ens permetrà seleccionar a quina alçada volem que treballi per tal que el pla  $z$  de treball tingui l'abast requerit.

### 4.3 Cinemàtica inversa

Per tal d'aconseguir que el robot es mogui tal i com es desitja cal definir unes equacions cinemàtiques. La part del robot que es vol controlar és l'efector final de la part mòbil i la part controlable són els angles que giren cadascun dels motors, per tant estem en el cas definit anteriorment com a cinemàtica inversa.

La forma de calcular l'angle de gir de cadascun dels motors és la mateixa per tots

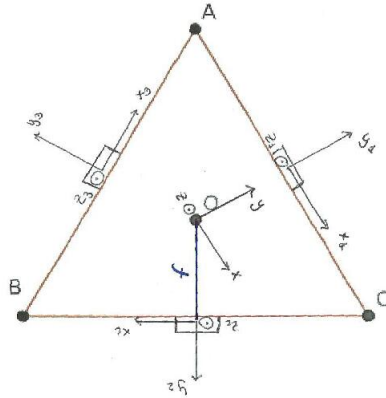


Figura 4: Canvi de Base

tres, l'únic que caldrà realitzar és un canvi de base posterior per a cada motor. Si definim un eix de coordenades centrat al triangle format pels tres motors tal i com es veu a la figura 4 realitzarem els següents canvis de base. La variable  $e$  és la homologa a la variable  $f$  però al triangle inferior.

$$[x_1, y_1, z_1] = [x_0, y_0 + e - f, z_0] \quad (1)$$

$$[x_2, y_2, z_2] = [-x_0 \cos(60) - y_0 \sin(60), x_0 \sin(60) - y_0 \cos(60) + e - f, z_0] \quad (2)$$

$$[x_3, y_3, z_3] = [-x_0 \cos(60) + y_0 \sin(60), -x_0 \sin(60) - y_0 \cos(60) + e - f, z_0] \quad (3)$$

Per tal de calcular l'angle girat per a cadascun dels motors utilitzem les següents equacions obtingudes al observar que la barra de mida  $rf$  només pot fer una trajectòria circular i que la barra de mida  $re$  realitza una trajectòria esfèrica, tal i com es veu en la figura 5. El nostre objectiu és calcular l'angle  $\theta$  i les dades que tenim són:

$$rf, re, x_f, y_f, z_f \quad (4)$$

A partir de les equacions típiques del cercle i de l'esfera obtenim les següents condicions.

$$x = 0 \quad (5)$$

$$z^2 + y^2 = rf^2 \quad (6)$$

$$(x - x_f)^2 + (y - y_f)^2 + (z - z_f)^2 = re^2 \quad (7)$$

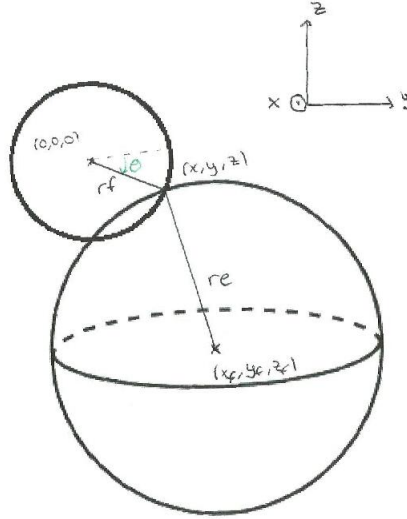


Figura 5: Càlcul de l'angle

Tenim un sistema de 3 variables i 3 equacions que podem resoldre.

$$x_f^2 + y^2 - 2yy_f + y_f^2 + z^2 - 2zz_f + z_f^2 = r_e^2 \quad (8)$$

$$-2yy_f - 2zz_f = r_e^2 - x_f^2 - y^2 - y_f^2 - z^2 - z_f^2 \quad (9)$$

Per no anar arrossegant tants valors direm que:

$$r_e^2 - x_f^2 - y^2 - y_f^2 - z^2 - z_f^2 = k \quad (10)$$

Utilitzant l'equació 6 obtenim:

$$2z_f \sqrt{r_f^2 - y^2} = k + 2yy_f \quad (11)$$

$$y^2(z_f^2 + y_f^2) + yy_f k + \frac{k^2}{4} - r_f^2 z_f^2 = 0 \quad (12)$$

$$y = \frac{-y_f k \pm \sqrt{y_f^2 k^2 - 4(z_f^2 + y_f^2)(\frac{k^2}{4} - r_f^2 z_f^2)}}{2(z_f^2 + y_f^2)} \quad (13)$$

$$z = \pm \sqrt{r_f^2 - y^2} \quad (14)$$

Un cop obtinguts els valors de x, y i z podem trobar el valor de theta.

$$\theta = \arctan\left(\frac{z}{y}\right) \quad (15)$$

Un cop s'ha obtingut l'angle pel primer dels motors simplement caldrà de realitzar els canvis de base anteriors per obtenir els angles que han de girar els altres dos motors.[6]

## 5 El robot facilitat

A la figura 6 es pot observar el robot amb tots els seus components que s'aniran descrivint en els següents subapartats. El robot consta de diferents tipus de components, que es poden agrupar en els que conformen l'estructura del robot, la part electrònica del robot i el suport del robot.

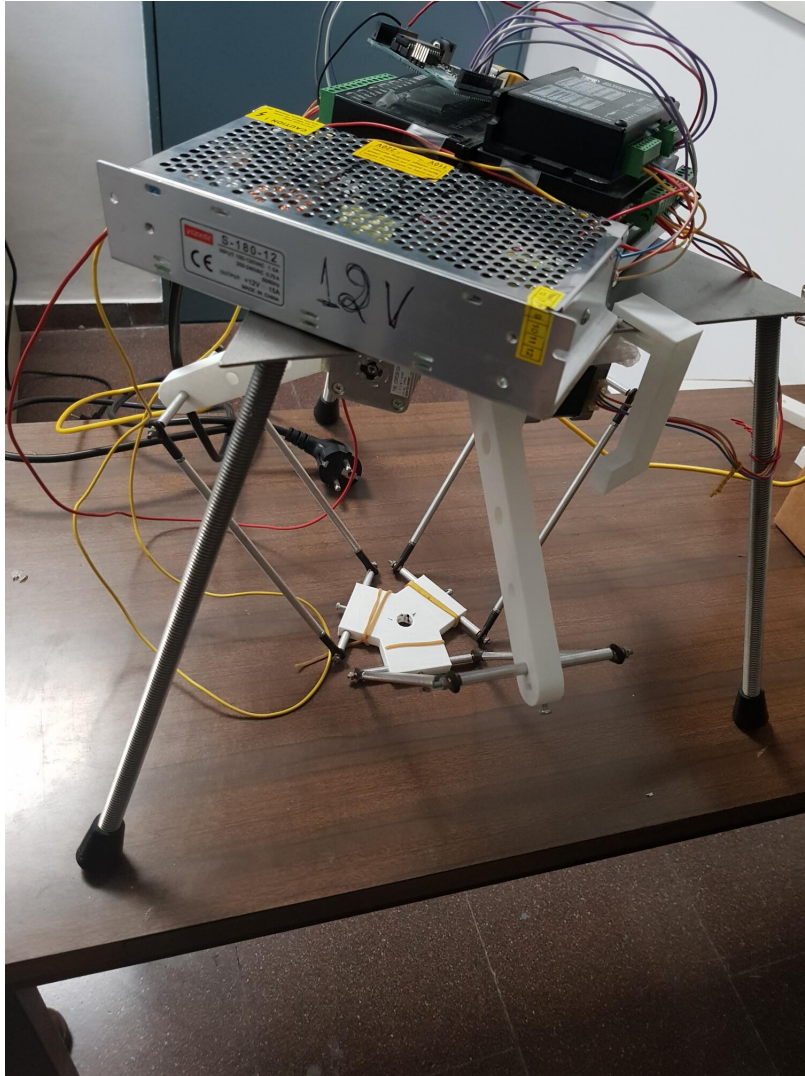


Figura 6: Robot

## 5.1 Base fixa

La base fixa del robot facilitat no té forma triangular, tal i com es pot veure a la figura 7, això no suposa cap inconvenient ja que els motors segueixen formant un angle de  $120^\circ$  entre ells. Al no tenir forma de triangle equilàter cal redefinir la variable 'f', que generalment es considera com la mida del costat de la base fixa, en aquest cas la variable 'f' serà la distància entre el motor i el centre de la base fixa. La distància 'f' és de 40mm. Aquesta base fixa va ser dissenyada mitjançant un programa CAD i va ser feta mitjançant una impressora 3D. La base té unes sortides verticals, és a aquestes sortides a les quals els motors es col·loquen mitjançant cargols i a on aquests s'uneixen amb els braços corresponents.



Figura 7: Base fixa

## 5.2 Base mòbil

La base mòbil tampoc té forma triangular, tal i com es pot veure a la figura 8 tot i que s'intueix degut a que les barres amb les quals està unida si que en formen un. Per tal de mantenir una coherència general en les variables, la variable 'e', que generalment es considera com la mida del costat de la base mòbil, es defineix com a la distància entre el mig del costat de la base mòbil fins el centre de la base, punt on es col·locarà l'actuador. La distància 'e' és de 45mm. Aquesta base també va ser dissenyada amb CAD i feta amb una impressora 3D. La base mòbil té un forat de diàmetre=10mm ja que allà és on s'ha de col·locar l'actuador del qual es parlarà a continuació. Aquesta base va unida als avantbraços mitjançant unes barres de metall, es pot observar que hi ha un cargol que sobresurt a la meitat de cada unió, la funció d'aquest cargol és de millorar la unió i evitar que es pogués produir joc en aquesta unió, cosa que seria una motiu d'imprecisions.

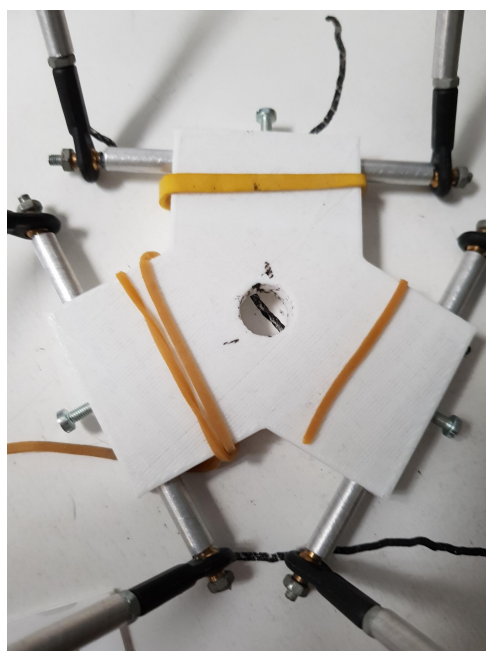


Figura 8: Base movil



### 5.3 Braç

El braç té la forma que es pot veure a la figura 9 serveix per a connectar el motor amb l'avantbraç i la seva mida correspon a la variable 'rf'. En aquest cas la mida és de 150mm. Aquesta peça també va ser dissenyada amb CAD i feta amb una impressora 3D. Es pot observar que hi ha una sèrie de forats, aquests forats estan pensats per poder modificar la longitud de la variable "rf" sense la necessitat d'utilitzar una peça completament nova, per modificar la longitud només caldria desmuntar una part de l'avantbraç i introduir la barra que connecta braç i avantbraç al forat que correspon a la longitud desitjada, aquesta barra es pot observar a la figura 10.



Figura 9: Braç

## 5.4 Avantbraç

L'avantbraç té la forma que es pot observar a la figura 10, en aquest cas la forma és diferent a la resta ja que es conforma amb dues barres que estan unides entre elles i formant una estructura rectangular. Aquesta forma aporta més rigidesa i estabilitat això permet, per tant, resistir esforços majors. És important que l'avantbraç elimini les possibles vibracions produïdes, aquestes es poden produir tant pels motors com per l'impacte produït al agafar un objecte, si no s'eliminessin la precisió i la vida útil del robot es veurien reduïdes. La variable que correspon a la mida de l'avantbraç és 're'. En aquest cas la mida és de 200mm. En les unions entre les barres cal permetre dues rotacions, la primera sobre l'eix que representarien les barres que van unides tant al braç com a la base mòbil i la segona sobre l'eix que seria la normal al pla format pel conjunt de les barres, per tal de permetre aquestes rotacions s'utilitza la peça de plàstic que és una ròtula esfèrica.



Figura 10: Avantbraç

## 5.5 Actuador

Per tal de poder adaptar el robot a la funcionalitat que se li vulgui donar, a la base mòbil hi ha un forat de diàmetre 1cm, aquest forat es veu a la figura 7. Amb una impressora 3D es pot realitzar el disseny de l'actuador que es vulgui utilitzar. Podem trobar diferents tipus d'actuadors per a robots Delta, per el que estan pensats per a realitzar la funció "pick and place" el més comú és tenir algun tipus de pinça o algun sistema que se li assimili, com molt bé podria ser un electroimant. En el cas particular d'aquest robot s'ha decidit optar per una peça que faci d'una espècie de pala i que sigui capaç d'agafar la peça mitjançant la pròpia inèrcia de la peça i del robot sense la necessitat de ser controlat amb una senyal extra com en el cas de les pinces o els electroimants. El disseny de l'actuador s'explica en un apartat posterior.

## 5.6 Motors

Els motors són els NEMA 16, la taula d'especificacions del fabricant es pot veure a la figura 11. Aquests són uns motors de bipolars del tipus pas a pas que treballen a 12 Vdc. El full de dades subministrat pel fabricant es pot trobar a l'annex.

Els motors pas a pas són actuadors electromagnètics rotacionals que converteixen

### Electrical Specification

Manufacturer Part Number: 16HS13-0404S  
 Motor Type: Bipolar Stepper  
 Step Angle: 1.8 deg  
 Holding Torque: 21Ncm(29.7oz.in)  
 Rated Current/phase: 0.4A  
 Voltage: 12V  
 Phase Resistance: 30ohms  
 Inductance : 32mH  $\pm$  20%(1KHz)

### Physical Specification

Frame Size: 39 x 39mm  
 Body Length: 34mm  
 Shaft Diameter:  $\Phi$ 5mm  
 Shaft Length: 22mm  
 Number of Leads: 4  
 Lead Length: 300mm  
 Weight : 190g

Figura 11: Taula d'especificacions dels motors

senyals digitals en moviments mecànics amb que tenen un desplaçament angular fixe. La posició de rotació depèn del número d'impulsos i la velocitat de la freqüència amb la que es reben. Una característica important en aquests tipus de motors és l'anomenat angle de pas, en el cas d'aquest tipus de motor és de  $1,8^\circ$ , l'angle de pas són els graus que girarà el motor al rebre un impuls.

Els motors pas a pas estan formats per un rotor sobre el qual s'integren diversos imants permanents i per un conjunt de bobines a l'estator. L'eix del motor està acoblat mecànicament sobre coixinets, cosa que permet que giri lliurement. El funcionament es basa en les forces que apareixen entre els camps magnètics dels imants permanents i els camps electromagnètics que es generen quan circula corrent elèctric; la bobina circular de l'estator es manté en una posició fixe i en el seu interior es troba la bobina del rotor que sota la influència d'aquests camps genera un parell electromagnètic que proporciona la rotació del motor com es pot veure en la figura 12.

Al polaritzar la bobina de l'estator, es creen els pols nord i sud del camp magnètic, aleshores el rotor intentarà seguir el moviment d'aquest camp i, per tant, orientarà els seus pols nord i sud cap als pols sud i nord de l'estator respectivament. Un cop s'han aconseguit alinear els pols del rotor i l'estator, l'estator commuta la orientació dels seus pols i el rotor s'haurà de tornar a alinear. És així com s'aconsegueix passar d'energia elèctrica a mecànica. El número de passos que gira el rotor quan es produeix la commutació de la polaritat en les bobines de l'estator és el que es

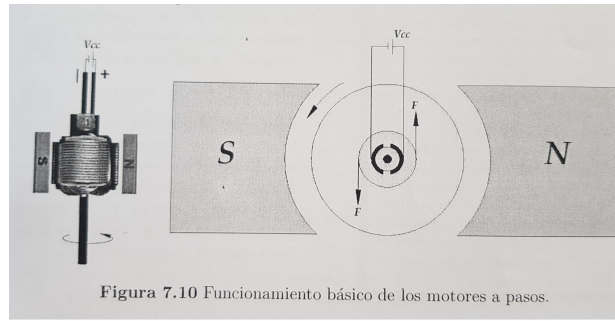


Figura 12: Imatge obtinguda del llibre d'Arduino[1]

denomina com a angle de pas.

Els motors pas a pas amb els quals s'està treballant són bipolars, això implica que estan formats per dues bobines i que en els extrems d'una bobina es poden tenir diverses polaritats, aleshores la corrent que circula per les bobines canvia de sentit en funció del voltatge. Del motor surten 4 cables que corresponen als dos extrems de cada bobina. Per aclariments mirar la figura 13. Els motors bipolars aporten l'avantatge que tenen aproximadament un 30 per cent més de parell motor que els unipolars, però el circuit és molt més complicat. Per connectar un motor bipolar a una placa Arduino cal introduir un dispositiu com els microcontroladors.[1]

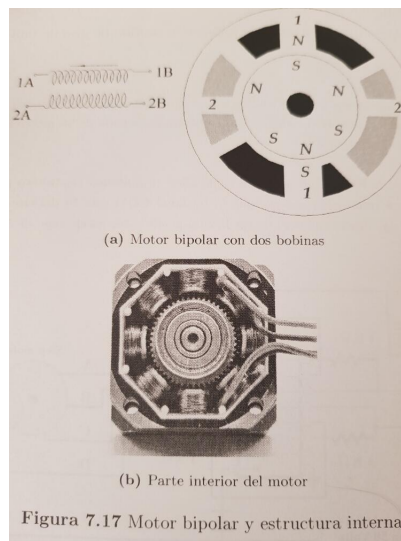


Figura 13: Imatge obtinguda del llibre d'Arduino[1]

## 5.7 Placa arduino

Arduino ofereix diferents tipus de plaques microcontroladors, en aquest cas s'ha treballat amb la placa Arduino Uno que es la que es pot veure a la figura 14. Compta amb 14 pins digitals d'entrada / sortida (dels quals 6 es poden utilitzar com a sortides PWM ("pulse width modulation")), 6 entrades analògiques, un cristall de quars de 16 MHz, una connexió USB, un encès de potència, un encapçalament ICSP i un botó de restabliment. Per tal de fer-la funcionar només cal connectar-la a l'ordinador mitjançant un cable USB. Els pins d'interès pel robot són els de tipus PWM, ja que són els que permeten controlar els motors, aquests pins són els que es connecten als microstep drivers per a poder realitzar la funció de control del robot.

Arduino a més ofereix una IDE ("integrated development environment") de codi obert anomenat Arduino Software que permet programar i pujar els programes de forma fàcil les seves plaques.



Figura 14: Foto de la placa Arduino Uno

## 5.8 Microstep drivers

Al treballar amb motor bipolars es precisa d'un element conegut com a microstep driver per a cadascun dels motors. La funció d'aquest element és principalment la de separar la potència que impulsa els motors de la potència que es troba a l'arduino. Altres funcions són la d'eliminar possibles vibracions ja que poden fraccionar els passos i al ser un sistema de control més precís es poden obtenir velocitats de treball més elevades.

Els microstep drivers amb els quals s'ha treballat són 2 de diferents, el primer és el 2H microstep driver DQ420MA que es pot veure a la figura 15, del qual només es té una unitat i per tant només controla un motor i el segon és el HY-DIV268N-5A, que es pot veure a la figura 16, que al tenir-ne dos unitats controlava dos motors. El manual dels controladors es pot trobar a l'annex.

El connexionat dels controladors consistia en connectar els cables de sortida del motor a les entrades A+, A-, B+ i B-, cada lletra consisteix en una de les bobines dels motors i el + i el - indiquen la polaritat. La resta de les entrades positives, van connectades als pins corresponents de la placa arduino, el pin que es connecta amb l'entrada DIR servirà per determinar la direcció de funcionament del motor, el de l'entrada PUL per determinar indicar quan ha d'avançar el motor, quan se li dona una pulsació i l'entrada EN per a determinar si el motor està actiu o no. Les entrades negatives estan connectades al terra elèctric.

Per triar el funcionament dels microstep drivers s'utilitzen els interruptors DIP, que permeten establir el número de passos que fa el motor per a realitzar una volta sencera i la corrent de sortida. Per obtenir una precisió més alta es va decidir utilitzar la configuració de 8 o 1/8 depenent del microstep driver que implica que podem emetre 1600 polsos per revolució.

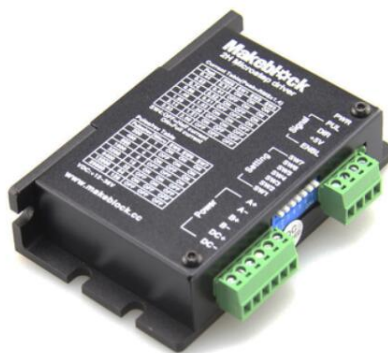


Figura 15: 2H microstep driver DQ420MA



Figura 16: Microstep driver HY-DIV268N-5A



## 5.9 Font de voltatge

Com s'especifica a la figura 11, els motors treballen a 12Vdc, per tant cal utilitzar un transformador que subministri els 12Vdc al sistema a partir de la corrent de la xarxa. La font de voltatge trobada es pot veure a la figura 17 i la seves especificacions les podem veure a la figura 18.



Figura 17: Font de voltatge

### Specification:

AC input voltage range: 85-264VAC 47-63Hz  
Input current: 110V /1.5A, 230V/0.75A  
Output: 12V/ 0-15A  
AC inrush current: Cold-start current 20A/115A 40A/230V  
Leakage current: <1mA/240VAC  
Dc adjustable range: +/-10% rated output voltage  
Overload protection: 105%-135% hiccup mode, auto-recovery  
Over-voltage protection: 115%-135% cut off the output, auto-recovery  
Setup, rise, hold up time: 200ms, 50ms, 20ms.  
Load regulation: +/- 1%  
Working temperature: -10°C~+60°C, 20%~90RH  
Weight/size: 0.58kg, 199x 97x 42mm

Figura 18: Especificacions

## 5.10 Suport del robot

El suport del robot és una estructura metàl·lica que consta de dues parts, les 3 potes, que tenen un tac per tal de protegir la taula de treball i una plataforma superior de forma triangular. A sobre de la plataforma triangular s'han col·locat la major part dels dispositius electrònics per tal d'evitar que el cablejat es mogués de forma excessiva. Aquesta plataforma està unida a la base fixa del robot mitjançant un cargol i una femella, que es poden observar a la figura 7. L'alçada del suport és de 305mm, a aquesta alçada cal restar el gruix de la base fixa i dels motors per obtenir l'alçada de treball del robot que és de 260mm.

## 6 Disseny de l'actuador

L'actuador que s'ha decidit col·locar, com ja s'ha dit prèviament, és un del tipus "pala", que permet agafar i deixar anar la peça amb la pròpia inèrcia del moviment del robot i de la peça en la cinta transportadora. Els altres sistemes es van descartar per la necessitat de controlar-los, ja que per tal d'activar l'electroimant o la pinça es requereix d'un senyal que s'ha de sincronitzar amb el moviment de tot el robot.

Per tal de fer el disseny de l'actuador s'ha de decidir quina peça serà la que el robot agafarà, s'ha optat per una peça d'escacs tal i com es pot veure a la figura 19, més concretament l'alfil, ja que té unes dimensions adequades pel sistema. La peça s'agafa pel punt on és més estreta que està a una alçada aproximada d'entre 2,5 i 3cm, aquest punt té una amplada aproximada de 1 cm, l'anell de sobre d'aquest punt té un diàmetre aproximat de 2cm, cosa que permet que la peça s'aganti amb major facilitat. La peça té un pes de 10 grams i per tant el robot la pot aixecar amb bastanta facilitat.

Tenint en compte les condicions de la peça i de la base mòbil s'ha dissenyat la



Figura 19: Peça triada

següent peça amb el programa SolidWorks i s'ha imprès amb una impressora 3D. El disseny de la peça es pot observar a la figura 20, les vistes amb les cotes es poden observar a les figures 21a, 21b i 21c. L'entrada que té la peça té una amplada de 1,5cm, que és prou ample per que la peça entri bé i prou estreta per que l'amplada de l'anell superior de la peça d'escacs serveixi com a sistema per a que aquesta no caigui. La base de l'actuador té una alçada de 5mm que no provoca cap problemàtica a l'hora d'agafar la peça. Les cantonades de la base estan arrodonides per facilitar l'entrada de la peça al moment d'agafar-la. L'alçada total de la peça és de 5,5cm on 0,5cm són de la base i els altres 5 són de la columna que ens permetrà unir la base

amb la base mòbil, en aquests 5 cm n'hi han 2 de rosca que ens permetran subjectar l'actuator amb la base mòbil i regular l'alçada, aquesta haurà de ser d'un mínim de 3cm ja que aquesta és la distància que hi ha entre l'anell i el punt més elevat de la peça d'escacs.

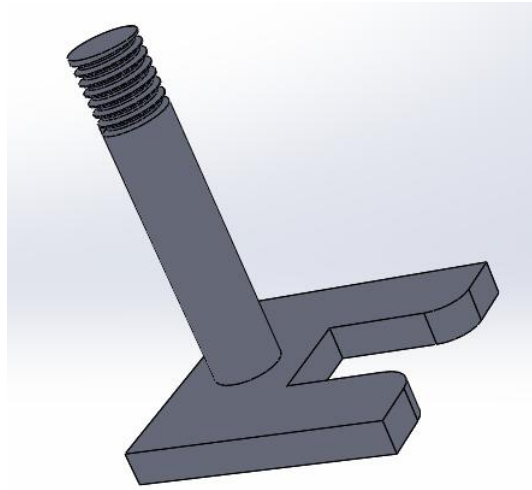


Figura 20: Actuator

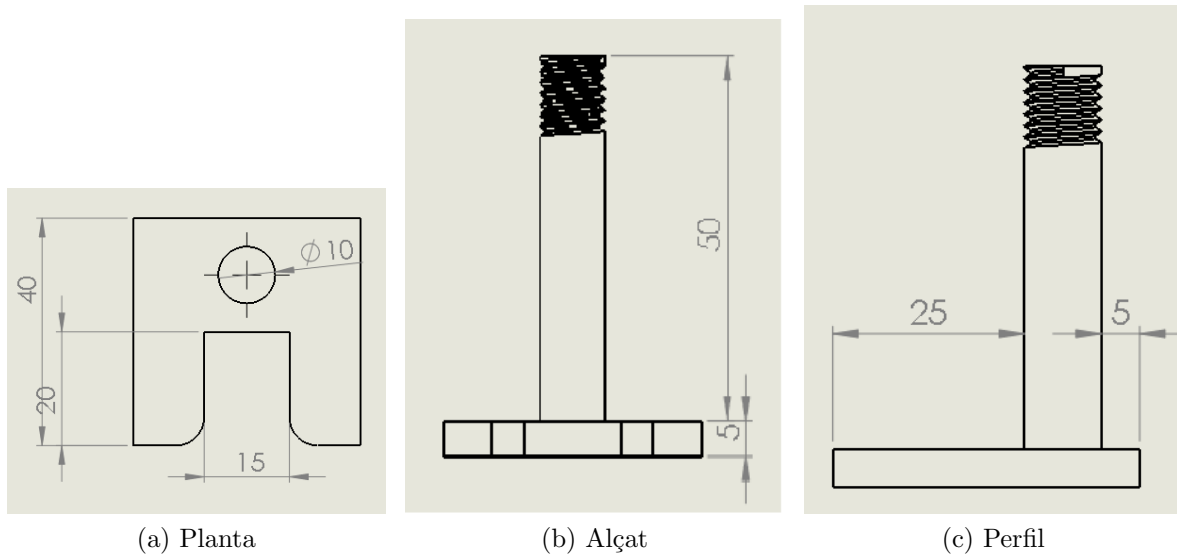


Figura 21: Vistes de la figura dissenyada

## 7 Abast

Per conèixer l'abast del robot cal fer una simulació de les posicions assolibles pel diferents angles de gir de cadascun dels motors. Al voler determinar les diferents posicions a través dels angles s'utilitza la cinemàtica directa. Mitjançant el codi de C++ s'han pogut obtenir les posicions on el robot pot arribar si els motors giren de  $0^\circ$  a  $90^\circ$  amb un increment de  $1^\circ$ . Els motors, teòricament, poden realitzar girs de fins a  $360^\circ$ , però per els motius que s'expliquen a continuació es limita a  $90^\circ$ . El rang de gir dels motors triats és de  $0^\circ$  a  $90^\circ$  ja que són els girs que permet realitzar el programa sense produir errors de càlcul, ja que amb angles majors a vegades el colze va cap a dalt i això és una posició no lògica, les posicions reals obtingudes per a girs de  $-90^\circ$  a  $0^\circ$  serien duplicats de les posicions obtingudes amb el càlcul realitzat. No es poden assolir girs de molt més de  $180^\circ$ , ja que de produir-se, provocarien un xoc entre el braç i la plataforma que aguanta el robot, per tant, el motor quedaria bloquejat i podria fer-se malbé.

Amb aquest codi s'ha obtingut una matriu de dades amb les posicions XYZ que el robot pot assolir en la situació real. Aquestes dades s'han passat a un Excel amb el qual s'han treure una sèrie de gràfics. Amb el programa XLSTAT, és un programa que s'acobla a Excel i que permet obtenir gràfiques en 3 dimensions, s'han pogut treure les següents figures 22, 23 i 24.

L'origen de coordenades dels eixos es correspon a la posició (0,0,-100). El pla

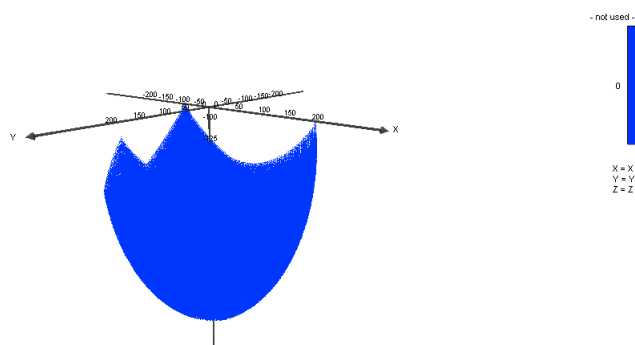


Figura 22: Panoràmica: Punts als quals es pot situar l'efector del robot

$z=0\text{mm}$  és el que formen els 3 motors, es prenen els valors de  $z$  que es troben per sota d'aquest pla com a valors negatius, així que quan es parla del pla  $z=-100\text{mm}$ , s'està parlant del pla que es troba  $100\text{mm}$  per sota dels motors.

Es pot observar que l'abast del robot en el pla  $z$  més allunyat dels motors és d'un sol punt, aquest punt correspon a la posició de l'efector quan tots 3 motors han girat  $90^\circ$ , la posició és la (0,0,-350). Al anar augmentant el valor del pla  $z$  es pot observar com aquest cada cop es va fent més ampli. A la figura de l'alçat, figura 23, es pot veure que s'obté una distribució parabòlica dels punts d'abast del robot. Fins arribar a un pla que es determinarà a continuació. Si s'hagués fet la simulació sense la limitació dels angles de gir dels motors, aquesta paràbola s'hagués anat tancant

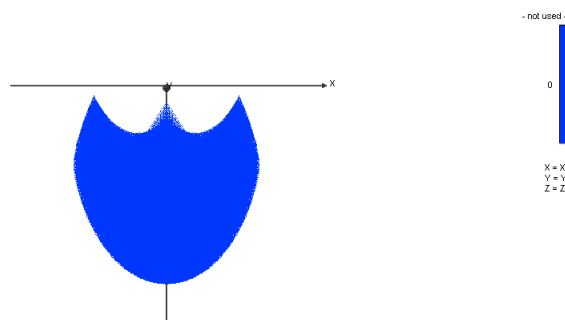


Figura 23: Alçat: Punts als quals es pot situar l'efector del robot

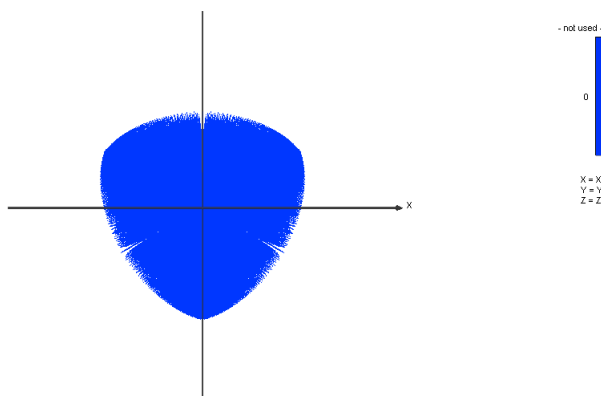


Figura 24: Planta: Punts als quals es pot situar l'efector del robot

fins arribar al punt  $(0,0,-50)$ , corresponent als angles de gir de  $270^\circ$  dels 3 motors.

Un altre aspecte a considerar és la quantitat de punts d'abast que podem trobar en cada pla  $z$ , es pot observar que al anar del pla més allunyat fins a un pla que es determinarà posteriorment i que coincideix amb el pla mencionat anteriorment, aquesta quantitat de posicions assolibles del robot va en augment, però al superar aquest pla comença a disminuir ja que hi ha posicions no assolibles i es formen uns espais buits a la figura. Aquest aspecte es pot observar en les vistes de la panoràmica com de l'alçat que correspon a les figures 22 23. Si s'hagués realitzat la simulació sense limitar els girs dels motors, s'hagués obtingut una major quantitat de punts assolibles dintre de l'espai considerat.

En la figura corresponent a la planta, figura 24 es poden observar unes línies buides que correspondrien a les posicions dels motors projectades als plans corresponents. Aquestes línies són posicions no assolibles pel robot.

A partir d'aquest estudi es va procedir a determinar quin era el millor pla de treball per el robot. Com que es parla d'una cinta transportadora l'objectiu és trobar el pla XY més ampli possible. Per tal d'obtenir el valor de la coordenada  $z$  per la qual el robot pot arribar a unes posicions XY més amples es van mirar diversos valors de  $z$  constant per tal d'anar acotant el pla desitjat. En les figures 25, 26 i 27, podem

veure diferents possibilitats de plans XY de treball.

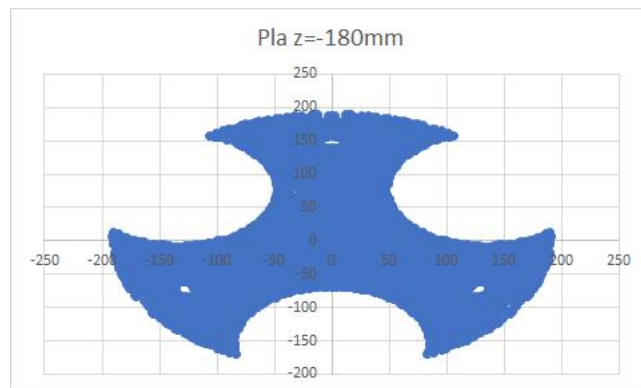


Figura 25: Punts d'abast en el pla  $z=-180\text{mm}$

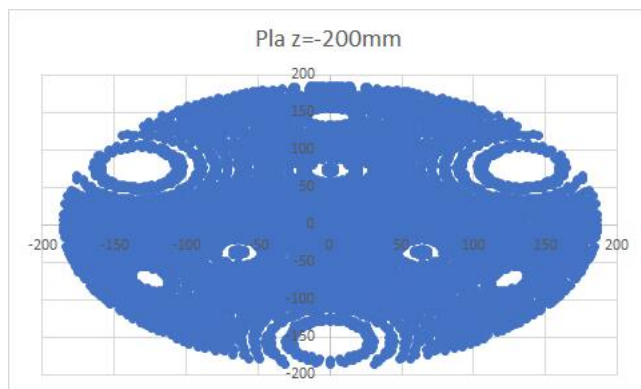


Figura 26: Punts d'abast en el pla  $z=-200\text{mm}$

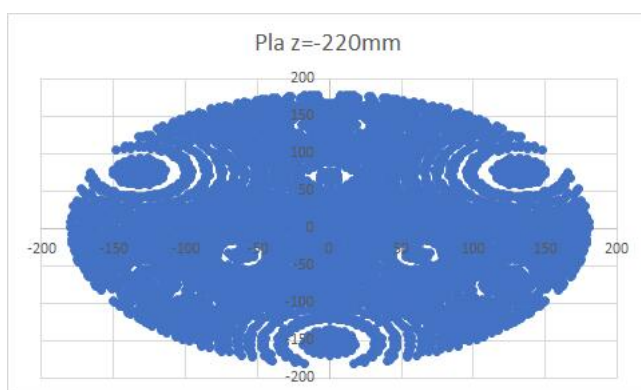


Figura 27: Punts d'abast en el pla  $z=-220\text{mm}$

Per tal de determinar amb més facilitat quin pla és el més adequat es va realitzar una adaptació del codi que permetia obtenir les posicions corresponents als valors de la coordenada x màxima i mínima i les posicions corresponents als valors de la

coordenada y màxima i mínima. Aquests valors es poden veure a la taula 1.

Pla z	Posició amb x màx	Posició amb x mín	Posició amb y màx	Posició amb y mín
-180	(191, 6, -180)	(-191, 6, -180)	(17, 189, -180)	(83, -171, -180)
-200	(186, 0, -200)	(-186, 0, -200)	(0, 186, -200)	(0, -186, -200)
-220	(180, 1, -220)	(-180, 1, -220)	(8, 179, -220)	(17, -181, -220)

Taula 1: Taula posicions

Al veure les característiques dels plans es va optar per treballar en el pla  $z=-200\text{mm}$ . Aquest pla és el que ofereix un major rang de treball i una major concentració de posicions assolibles. Per tal de treballar en aquest pla caldrà tenir en compte la mida de l'actuador.



## 8 Trajectòries

### 8.1 Definició i paràmetres

Una trajectòria és la línia que descriu un mòbil en l'espai", aquesta es la definició que es pot trobar al Diccionari de la llengua catalana. En el nostre cas la trajectòria que és objecte d'estudi és la que ha de realitzar l'efector del robot. Una trajectòria be definida per un punt inicial i un punt final que s'han d'unir mitjançant un camí realitzable. Altres paràmetres importants són la velocitat a la qual es realitza el camí i l'acceleració que tenim a cada punt.

## 8.2 Algoritmes

El problema bàsic que apareix al planificar el moviment dels robots consisteix en que, donades dues posicions, una inicial i una de final, cal trobar un camí lliure de col·lisions que uneixi aquestes dues posicions. Al treballar amb un espai lliure de objectes i estacionari i coneixent l'abast no cal preocupar-se de possibles colisions amb altres objectes. Hi ha diferents tipus d'algoritmes per a resoldre aquesta problemàtica que es poden classificar de la següent manera segons quin tipus de solució donen.

Els algoritmes poden ser complets o incomplets. Un algoritme complet són els que garanteixen la recerca d'un camí al problema sempre que aquest camí existeixi, si el camí no existeix retornaran que no hi ha cap camí que resolgui el problema. Un algoritme incomplet no garantirà la recerca d'un camí.

Els algoritmes de Resolució-Complets, treballen amb una resolució reduïda. Es discretitza l'espai i es fa una recerca completa d'un camí en aquest espai.

Els algoritmes Probabilísticament Complets, són aquells que garanteixen trobar un camí si aquest existeix realment.

Els algoritmes Heurístics, són aquells on es permeten configuracions impossibles per a facilitar els càlculs, ja que els algoritmes complets són, generalment, computacionalment inviables.

Els algoritmes es poden classificar també en globals o locals, depenent de l'espai de treball que s'utilitzi a l'hora d'aplicar-lo. Si s'utilitza tot l'espai existent es tracta d'un algoritme global, mentre que si s'utilitza un espai reduït es tracta d'un algoritme local.[7]

Els mètodes per a la resolució del problema de la planificació es poden classificar en els següents. Mètode del camp potencial, mètode de la descomposició en cel·les i mètode dels mapes de camins.

El mètode del camp potencial consisteix en definir un potencial repulsiu als obstacles i un potencial atractiu a la destinació final. El potencial total és la combinació lineal de tots els potencials atractius i repulsius definits. El problema d'aquest mètode és que s'acaben introduint mínims locals que condueixen a posicions de bloqueig del robot, per treure el robot d'aquestes posicions cal fer una presa de decisions basades en una posició particular i per tant es limita les opcions d'aplicar aquest mètode.

El mètode de la descomposició en cel·les és el conjunt de l'aplicació de diversos mètodes, que es basen en la descomposició de les configuracions lliures en un conjunt de regions anomenades cel·les. A partir d'aquestes cel·les es representa un graf de connectivitat de l'espai lliure. Per tant simplement s'ha de resoldre el problema de trobar un camí en el graf que connecti els punts inicial i final.

El mètode de mapes de camins es basa en la construcció prèvia de la connectivitat

de l'espai lliure mitjançant unes corbes conegudes com a camins. Un cop realitzat aquest mapa de corbes cal trobar aquell camí que connecta el punt inicial amb el final.

Una trajectòria és la línia que descriu un mòbil en l'espai", aquesta es la definició que es pot trobar al Diccionari de la llengua catalana. En el nostre cas la trajectòria que és objecte d'estudi és la que ha de realitzar l'efector del robot.

### 8.3 Trajectòria realitzada

Al realitzar-se sempre la mateixa trajectòria, ja que es tracta d'un robot "pick and place" que treballa amb peces que sempre passen per la mateixa posició i amb una freqüència constant, s'ha programat la trajectòria fixant els punts de treball als quals es vol que vagi el robot i traçant les línies més rectes possibles. Per tal de calcular la trajectòria s'ha utilitzat el programa de la cinemàtica inversa i directa fet en C++, i s'ha creat una taula amb Excel per trobar els valors del que ha de girar cada motor.

Per tal de que el robot es situï sempre en la mateixa posició inicial s'utilitzen tres subjeccions que surten de la plataforma superior del suport general del robot i que serveixen per subjectar cadascun dels braços del robot. Aquestes subjeccions es poden veure a les figures 28 i 29, les mides són les següents té una alçada de 8cm i l'amplada de la zona de subjecció amb la plataforma superior del suport és de 2,5 cm i l'amplada de la zona de subjecció del braç és de 1,5cm. Al col·locar les subjeccions la base fixa se situa al pla  $z=-140\text{mm}$  i l'efector en el punt  $(0,0,-140)$ .

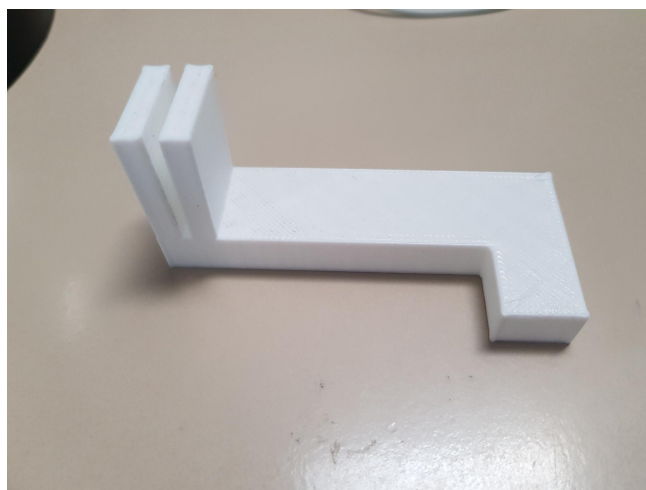


Figura 28: Subjecció utilitzada

La trajectòria consisteix en els següents passos:

- Baixar el robot fins al pla on es pot recollir la peça.
- Desplaçament en l'eix x fins a recollir la peça.
- Pujar fins a un pla on la peça no toqui el terra.
- Desplaçament en l'eix x fins arribar al punt on es deixarà la peça.
- Baixar el robot fins el pla on es deixa la peça.
- Pujada del robot sense la peça.
- Retorn a la posició inicial.

A la figura 30 es pot veure com s'aconsegueix arribar a aquestes trajectòries. A la primera columna podem trobar el pas que estem realitzant, es pot observar que

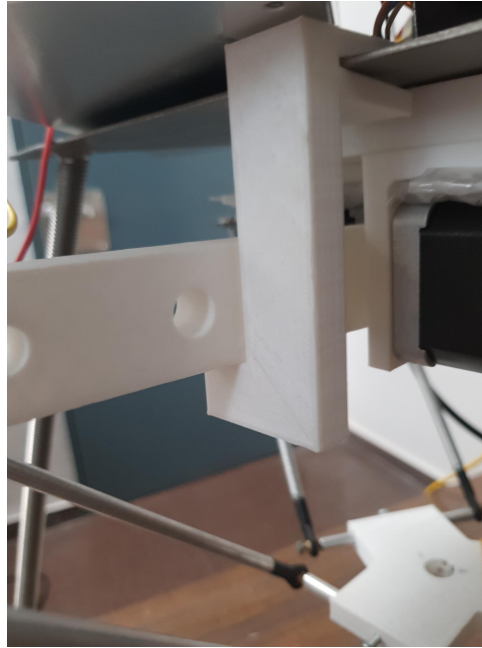


Figura 29: Subjecció col·locada en el robot

PAS	X	Y	Z	theta1	theta2	theta3	diff1	diff2	diff3	steps1	steps2	steps3
1	0,00	0,00	-140,00	2,28	2,28	2,28	-	'-	'-	'-	'-	'-
2	0,00	0,00	-175,00	14,01	14,01	14,01	-11,73	-11,73	-11,73	-6,52	-6,52	-6,52
2'	0,00	0,00	-177,80	14,88	14,88	14,88	-12,60	-12,60	-12,60	-7,00	-7,00	-7,00
3	-50,00	0,00	-180,02	17,68	30,76	3,59	-2,80	-15,88	11,29	-1,56	-8,82	6,27
3'	-50,13	0,82	-179,54	18,48	31,08	4,08	-3,60	-16,20	10,80	-2,00	-9,00	6,00
4	-50,00	0,00	-140,00	5,70	22,51	-11,57	12,78	8,57	15,65	7,10	4,76	8,69
4'	-51,10	-2,03	-142,44	5,88	23,88	-10,32	12,60	7,20	14,40	7,00	4,00	8,00
5	50,00	0,00	-142,44	6,53	-10,49	23,04	-0,65	34,37	-33,36	-0,36	19,10	-18,53
5'	51,59	1,04	-143,98	7,68	-10,32	23,88	-1,80	34,20	-34,20	-1,00	19,00	-19,00
6	50,00	0,00	-175,00	16,84	2,56	30,14	-9,17	-12,88	-6,26	-5,09	-7,16	-3,48
6'	50,13	0,82	-179,54	18,48	4,08	31,08	-10,80	-14,40	-7,20	-6,00	-8,00	-4,00
7	50,00	0,00	-140,00	5,70	-11,57	22,51	12,78	15,65	8,57	7,10	8,69	4,76
7'	51,10	-2,03	-142,44	5,88	-10,32	23,88	12,60	14,40	7,20	7,00	8,00	4,00
8	0,00	0,00	-140,00	2,28	2,28	2,28	3,60	-12,60	21,60	2,00	-7,00	12,00
8'	0,00	0,00	-140,00	2,28	2,28	2,28	3,60	-12,60	21,60	2,00	-7,00	12,00

Figura 30: Taula pel càlcul de la trajectòria

cada pas està calculat 2 cops excepte el pas 1, el primer dels 2 cops es calculen els girs que s'haurien de realitzar per arribar a la posició desitjada mitjançant el codi corresponent a la cinemàtica inversa. Aquestes diferències en els angles inicials i finals es poden veure en les columnes diff, en la majoria de casos aquests girs no es poden realitzar ja que al tractar-se de motors pas a pas només poden realitzar girs múltiples de  $1,8^\circ$  que és el seu angle de pas. A les columnes steps es poden veure els números de passos que s'han de realitzar per a fer el gir desitjat. El segon cop que es calcula el mateix pas es parteix del número de passos calculats el primer cop i s'agafa el valor enter d'aquest resultat, es multiplica per 1,8 i així s'obté la diferència real que hi haurà entre l'angle inicial i el final, mitjançant aquesta diferència i coneixent els angles inicials de cada motor es troben els angles finals, finalment s'utilitza el programa per a calcular la cinemàtica directa i saber a quina posició de l'efector corresponen els angles finals.

Un cop coneguts el número de passos que calen per arribar a cadascuna de les posi-

cions es van introduir les dades a un programa de arduino per repetir la trajectòria en bucle. Cal tenir en consideració que el punt de treball està uns 3cm per sota del pla on està l'efector, ja que és aquest on es troba l'actuador. S'ha decidit baixar fins al pla  $z=-175$  per recollir la peça ja que el terra està al pla  $z=-255$  i l'alçada de la peça d'escacs més l'actuador és de 8cm amb una tolerància de 1cm. Al treballar amb línies rectes la velocitat és constant i hi ha una aturada una de 5microsegons quan hi ha un canvi de direcció.

## 8.4 Altres possibilitats pel càlcul de trajectòries

Al fer recerca s'ha vist que es pot adaptar la plataforma arduino per a treballar amb g-code. G-code és el llenguatge de programació amb el qual treballen les màquines CNC, màquines de control numèric, aquest codi serveix per a donar instruccions de trajectòries, velocitats i acceleracions a màquines com torns, fresadores, impressores 3d...

Hi ha programes, com per exemple Inkscape, amb el qual s'ha treballat durant la carrera, que poden donar el G-code d'una trajectòria dibuixada. Amb aquest tipus de programa es podria obtenir el codi de la trajectòria que es vol realitzar en G-code i només caldria introduir-la a l'arduino. Per introduir la trajectòria en G-code a l'arduino s'ha trobat l'opció d'utilitzar la llibreria grbl, que és de codi obert i que permet que l'arduino interpreti G-code, un cop es puja el grbl a l'arduino cal un programa de control, es va trobar un anomenat ugs, "Universal G-code sender". Al realitzar les proves per aquest sistema es va veure que la llibreria grbl considera que cada motor és l'encarregat de controlar una direcció de treball, cosa que no és així en els robots delta i per tant no es va poder utilitzar.

## 9 Possibilitats de desenvolupament del treball

Com a possibles treballs futurs es poden trobar dos vies de desenvolupament. La primera és la de seguir desenvolupant la funcionalitat "pick and place", aquest desenvolupament es pot fer incloent sistemes de sensors que siguin capaços de determinar quan passarà l'objecte a agafar. Amb aquest nou sistema es podria solucionar el sistema del recolliment de objectes que no passin amb una freqüència constant. Però per aconseguir l'objectiu de poder canviar la posició per la qual passa el robot cal passar per una segona via de desenvolupament.

La segona via de desenvolupament consisteix en el desenvolupament d'un codi que permeti que el robot sigui capaç de situar la posició per on ha de passar la peça i generar ell mateix la trajectòria que ha de realitzar per a agafar-la.



## 10 Conclusions

Aquest treball ha demostrat la viabilitat de la realització d'un prototip de robot Delta per a una aplicació del tipus "pick and place". Per tal de veure aquesta viabilitat s'anirà comprovant la realització dels objectius proposats al començament del treball.

El primer objectiu consistia en la comprensió de la mecànica del robot. Aquest objectiu s'ha assolit al poder descriure la cinemàtica tant directa com inversa del robot. Obtenir les equacions de la cinemàtica era una condició bàsica per la realització del treball, ja que és a partir d'aquestes equacions que s'ha pogut fer l'estudi de l'abast del robot i la realització dels codis de càlcul de posicionament del robot.

La comprensió dels components electrònics i dels llenguatges de programació ha agilitzat els mètodes de càlcul i del desenvolupament del treball. Els motors bipolars pas a pas en combinació amb els microstep drivers juntament amb la placa arduino UNO han permès tot el desenvolupament electrònic i de control del robot.

La realització d'un sistema de control i posicionament de l'efector final s'ha realitzat de forma parcial. El robot és capaç de descriure una trajectòria programada manualment i que es repeteix. Al haver definit com a objectiu la realització d'un robot "pick and place" on els objectes a agafar passessin sempre per la mateixa posició i amb una freqüència constant, aquesta trajectòria única i repetitiva es dona per bona. Un sistema més desenvolupat queda a treballs futurs.

El robot es mou de forma suau i contínua al haver triat una bona configuració electrònica. La trajectòria a realitzar s'ha triat de forma correcta i al haver treballat amb línies rectes s'ha pogut treballar amb una velocitat constant.

## Referències

- [1] F. R. Cortés and J. C. Monjaraz, *Arduino: aplicaciones en robótica, mecatrónica e ingenierías*. Alfaomega, 2015.
- [2] I. Zabalza and J. Ros, “Aplicaciones actuales de los robots paralelos,” in *8th Latin American Congress of Mechanical Engineering. Cusco, Perú*, vol. 3, 2007.
- [3] R. Clavel, “Conception d’un robot parallèle rapide à 4 degrés de liberté,” 1991.
- [4] D. Wappling and S. Murphy, “Pick and place,” 2013. US Patent 8,565,912.
- [5] I. Iso, “Iso 10218-1: 2011: Robots and robotic devices—safety requirements for industrial robots—part 1: Robots,” *Geneva, Switzerland: International Organization for Standardization*, 2011.
- [6] E. Martínez, C. A. Peña, and E. Yime, “Diseño óptimo de un robot paralelo con configuración delta para aplicaciones educativas,” *Revista Educación en Ingeniería*, vol. 5, no. 10, pp. 110–119, 2010.
- [7] J. M. Bañón, O. Arcila, J. Arango, *et al.*, “Sobre los algoritmos de planificacion del movimiento de robots,” *Ingeniería y Competitividad*, vol. 2, no. 2, pp. 18–25, 2011.

## **11 Annex**

### **11.1 Normativa**

[5]

Mayo 2012

Versión corregida, Febrero 2014

### TÍTULO

**Robots y dispositivos robóticos**

**Requisitos de seguridad para robots industriales**

**Parte 1: Robots**

(ISO 10218-1:2011)

*Robots and robotic devices. Safety requirements for industrial robots. Part 1: Robots. (ISO 10218-1:2011)*

*Robots et dispositifs robotiques. Exigences de sécurité pour les robots industriels. Partie 1: Robots. (ISO 10218-1:2011)*

### CORRESPONDENCIA

Esta norma es la versión oficial, en español, de la Norma Europea EN ISO 10218-1:2011, que a su vez adopta la Norma Internacional ISO 10218-1:2011.

### OBSERVACIONES

Esta norma sustituye a la Norma EN ISO 10218-1:2008 (ratificada por AENOR).

### ANTECEDENTES

Esta norma ha sido elaborada por el comité técnico AEN/CTN 116 *Sistemas industriales automatizados* cuya Secretaría desempeña AER ATP.

## EXTRACTO DEL DOCUMENTO UNE-EN ISO 10218-1

# Índice

Prólogo .....	6
0 <b>Introducción</b> .....	7
1 <b>Objeto y campo de aplicación</b> .....	7
2 <b>Normas para consulta</b> .....	8
3 <b>Términos y definiciones</b> .....	8
4 <b>Identificación de peligros y evaluación de riesgos</b> .....	12
5 <b>Requisitos de diseño y medidas de protección</b> .....	13
5.1 <b>Generalidades</b> .....	13
5.2 <b>Requisitos generales</b> .....	13
5.3 <b>Controles de accionamiento</b> .....	14
5.4 <b>Funcionamiento del sistema de control de seguridad (hardware/software)</b> .....	14
5.5 <b>Funciones de paro del robot</b> .....	16
5.6 <b>Control de velocidad</b> .....	17
5.7 <b>Modos de funcionamiento</b> .....	18
5.8 <b>Controles de la consola de guiado</b> .....	19
5.9 <b>Control de movimientos simultáneos</b> .....	21
5.10 <b>Requisitos para el funcionamiento cooperativo</b> .....	22
5.11 <b>Protección de singularidades</b> .....	23
5.12 <b>Limitación de los ejes</b> .....	23
5.13 <b>Movimiento sin potencia de accionamiento</b> .....	25
5.14 <b>Disposiciones para la carga del robot</b> .....	25
5.15 <b>Conectores eléctricos</b> .....	25
6 <b>Verificación y validación de los requisitos de seguridad y las medidas de protección</b> .....	25
6.1 <b>General</b> .....	25
6.2 <b>Métodos de verificación y validación</b> .....	25
6.3 <b>Verificación y validación necesaria</b> .....	26
7 <b>Información de uso</b> .....	26
7.1 <b>Generalidades</b> .....	26
7.2 <b>Manual de instrucciones</b> .....	26
7.3 <b>Señalización</b> .....	28
<b>Anexo A (Informativo)      Lista de peligros significativos</b> .....	29
<b>Anexo B (Normativo)      Tiempos de parada y distancias métricas de parada</b> .....	35
<b>Anexo C (Informativo)      Características de las tres posiciones del dispositivo de validación</b> .....	37
<b>Anexo D (Informativo)      Características opcionales</b> .....	38
<b>Anexo E (Informativo)      Etiquetado</b> .....	40
<b>Anexo F (Normativo)      Medios para verificar los requisitos de seguridad y sus medidas</b> .....	41
<b>Bibliografía</b> .....	54

## 1 Objeto y campo de aplicación

Esta parte de la Norma ISO 10218 especifica los requisitos y las directrices para un diseño inherentemente seguro, las medidas de protección y la información para el uso de robots industriales. La norma describe los riesgos básicos asociados con los robots y proporciona los requisitos para eliminar o reducir adecuadamente los peligros asociados con estos riesgos.

Esta parte de la Norma ISO 10218 no trata el robot de forma completa. La emisión de ruido generalmente no está considerada un peligro significativo del propio robot, por lo que el ruido se excluye del objeto y campo de aplicación de esta parte de la Norma ISO 10218.

Esta parte de la Norma ISO 10218 no es aplicable a robots no industriales, aunque los principios de seguridad establecidos en la Norma ISO 10218 pueden ser utilizados para estos otros robots.

NOTA 1 Los ejemplos de aplicaciones de robots no industriales incluyen, pero no se limitan a, robots acuáticos, militares y espaciales, manipuladores teleoperados, prótesis y de ayuda para discapacitados físicos, micro-robots (con desplazamientos menores de 1 mm), robots de cirugía o médicos, de servicio y de productos al consumidor.

NOTA 2 Los requisitos para los sistemas robots, su integración e instalación se tratan en la Norma ISO 10218-2.

NOTA 3 Aplicaciones específicas pueden producir peligros adicionales (por ejemplo soldadura, corte por láser, mecanizado). Estos peligros asociados a sus respectivos sistemas deben considerarse en la fase de diseño del robot.

## 2 Normas para consulta

Las normas que a continuación se indican son indispensables para la aplicación de esta norma. Para las referencias con fecha, sólo se aplica la edición citada. Para las referencias sin fecha se aplica la última edición de la norma (incluyendo cualquier modificación de ésta).

ISO 9283:1998, *Robots manipuladores industriales. Criterios de análisis de prestaciones y métodos de ensayo relacionados.*

ISO 10218-2, *Robots y dispositivos robóticos. Requisitos de seguridad para robots industriales. Parte 2: Sistemas robot e integración.*

ISO 12100, *Seguridad de las máquinas. Conceptos básicos, principios generales para el diseño. Evaluación y reducción del riesgo.*

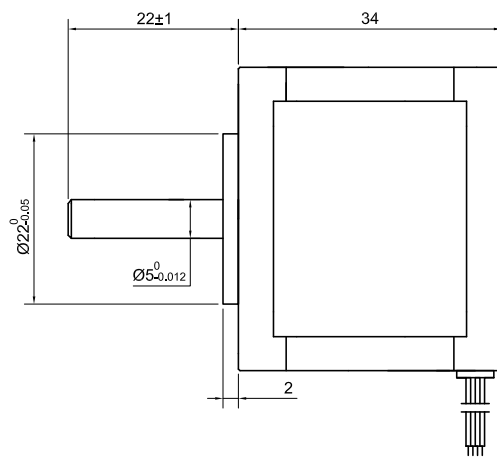
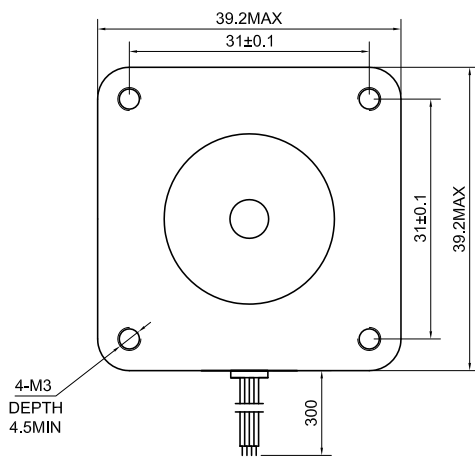
ISO 13849-1:2006, *Seguridad de las máquinas. Partes de los sistemas de mando relativas a la seguridad. Parte 1: Principios generales para el diseño.*

ISO 13850, *Seguridad de las máquinas. Parada de emergencia. Principios para el diseño.*

IEC 60204-1, *Seguridad de las máquinas. Equipo eléctrico de las máquinas. Parte 1: Requisitos generales.*

IEC 62061:2005, *Seguridad de las máquinas. Seguridad funcional de sistemas de mando eléctricos, electrónicos y electrónicos programables relativos a la seguridad.*

## 11.2 Fitxes tècniques

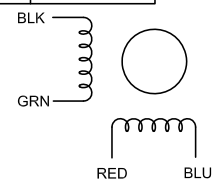


SPECIFICATION	CONNECTION	BIPOLAR
VOTAGE(VOC)		12.00
AMPS/PHASE		0.40
RESISTANCE/PHASE(Ohms)@25°C		30.00±10%
INDUCTANCE/PHASE(mH)@1KHz		32.00±20%
HOLDING TORQUE(Nm)[lb-in]		0.21[1.86]
STEP ANGLE(°)		1.80
STEP ACCURACY(NON-ACCUM)		±5.00%
ROTOR INERTIA(g-cm²)		20.00
WEIGHT(Kg)[lb]		0.18[0.40]
TEMPERATURE RISE:MAX.80°C (MOTOR STANDSTILL;FOR 2PHASE ENERGIZED )		
AMBIENT TEMPERATURE -10°C~50°C[14°F~122°F]		
INSULATION RESISTANCE 100 Mohm (UNDER NORMAL TEMPERATURE AND HUMIDITY )		
INSULATION CLASS B 130°C[266°F]		
DIELECTRIC STRENGTH 500VAC FOR 1MIN,(BETWEEN THE MOTOR COILS AND THE MOTOR CASE )		
AMBIENT HUMIDITY MAX.85%(NO CONDENSATION)		

TYPE OF CONNECTION (EXTERN)		MOTOR	
PIN NO	BIPOLAR	LEADS	WINDING
1	A —	BLK	A
2	A\ —	GRN	A\
3	B —	RED	B
4	B\ —	BLU	B\

FULL STEP 2 PHASE-Ex .  
WHEN FACING MOUNTING END (X)

STEP	A	B	A\	B\		
1	+	+	-	-	↓	CCW
2	-	+	+	-	↑	
3	-	-	+	+	↓	CW
4	+	-	-	+	↑	



**STEPPERONLINE**  
Motors & Electronics

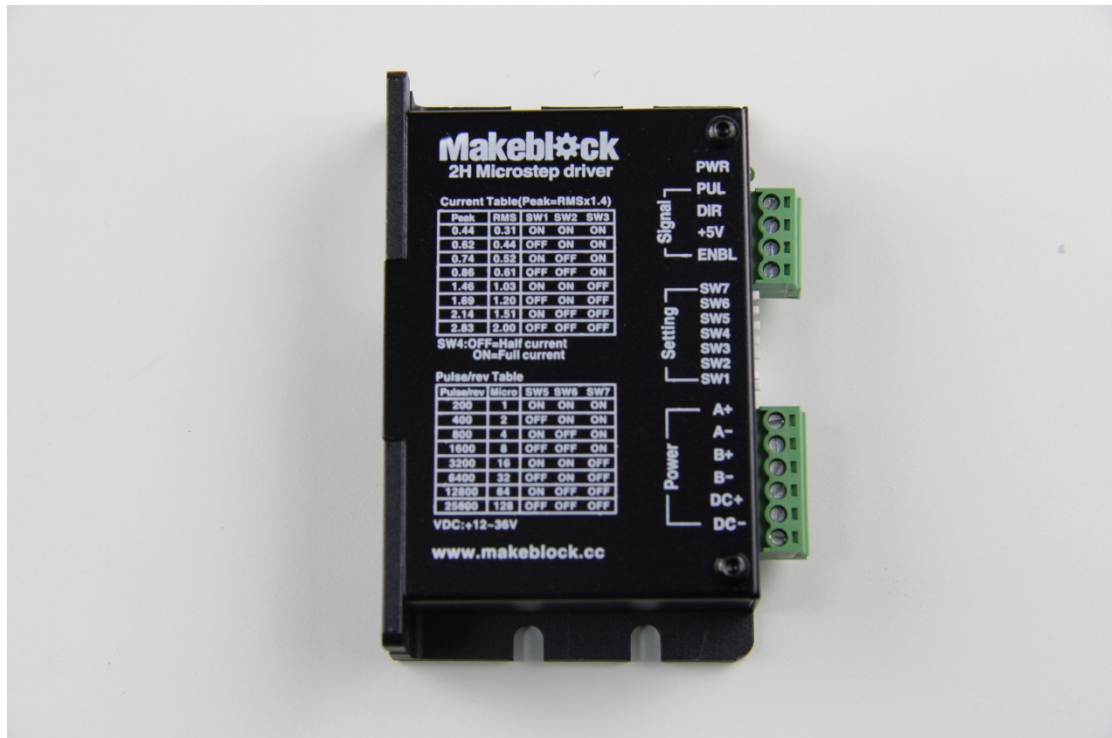
	APVD		
	CHKD		
1.5:1	DRN		
SCALE	SIGNATURE	DATE	

**STEPPER MOTOR**

16HS13-0404S



# Makeblock 2H Microstep Driver



## 1. Introduction:

Makeblock 2H Microstep Driver is a type of two-phase hybrid stepping motor driver, the drive voltage of which is from 12VDC to 36VDC. It is designed for use with 2-phase hybrid stepper motor of all kinds with 20mm to 42mm outside diameter and less than 2.0A phase current. This circuit that it adopts is similar to the circuit of servo control which enables the motor to run smoothly almost without noise and vibration. Holding torque when the Microstep Driver runs at high speed is also significantly higher than the other two-phase drivers, what's more, the positioning accuracy is also higher. It is widely used in middle and big size numerical control devices such as curving machine, CNC machine, and computer embroidery machine, packing machines and so on.

## 2. Features:

- High performance, low price
- Average current control, 2-phase sinusoidal output current drive
- Supply voltage from 12VDC to 36VDC
- Opto-isolated signal I/O
- Overvoltage, under voltage, overcurrent, phase short circuit protection

- 8 channels subdivision and automatic idle-current reduction
- 8 channels output phase current setting
- Offline command input terminal
- Motor torque is related to speed, but not related to step/revolution
- High start speed
- High holding torque at high speed

### 3. Electrical specification:

<b>Input voltage</b>	12-36VDC
<b>Input current</b>	< 2A
<b>Output current</b>	0.44A- 2.83A
<b>Consumption</b>	Consumption: 40W;
<b>Temperature</b>	Working Temperature -10~45°C; Stocking temperature -40°C~70°C
<b>Humidity</b>	Not condensation, no water droplets
<b>Gas</b>	Prohibition of combustible gases and conductive dust
<b>Weight</b>	70G

### 4. Pins assignments and description:

**Connector Pins Configurations Table:**

<b>Pin Function</b>	<b>Details</b>
PUL +,PUL-	Pulse signal: PUL+ is the positive end of pulses input pin PUL- is the negative end of pulse input pin
DIR+,DIR-	DIR signal: DIR+ is the positive end of direction input pin DIR- is the negative end of direction input pin
ENBL+	Enable signal: ENBL+ is the positive end of direction input pin. This signal is used for enabling/disabling the driver. High level for enabling the driver and low level for disabling the driver.
ENBL-	ENBL- is the negative end of direction input pin. Usually left unconnected (enabled)

## 5. Function choice (Using DIP pins to achieve this function)

### 5.1 Micro step resolution setting

Micro step resolution is set by SW 5,6,7,8 of the DIP switch as shown in the following

Micro step resolution

<b>SW5</b>	ON	OFF	ON	OFF	ON	OFF	ON	OFF
<b>SW6</b>	ON	ON	OFF	OFF	ON	ON	OFF	OFF
<b>SW7</b>	ON	ON	ON	ON	OFF	OFF	OFF	OFF
<b>Pulse/rev</b>	200	400	800	1600	3200	6400	12800	25600
<b>Micro</b>	1	2	4	8	16	32	64	128

### 5.2 Standstill current setting

SW4 is used for this purpose. OFF meaning that the standstill current is set to be half of the selected dynamic current and ON meaning that standstill is set to be the same as the selected dynamic current.

### 5.3 Output current setting

The first three bits (SW 1, 2, 3) of the DIP switch are used to set the dynamic current.

Select a setting

Closest to your motor's required current

<b>SW1</b>	<b>SW2</b>	<b>SW3</b>	<b>PEAK</b>	<b>RMS</b>
ON	ON	ON	0.44 A	0.31 A
OFF	ON	ON	0.62 A	0.44 A
ON	OFF	ON	0.74 A	0.52 A
OFF	OFF	ON	0.86 A	0.61 A
ON	ON	OFF	1.46 A	1.03 A
OFF	ON	OFF	1.69 A	1.20 A
ON	OFF	OFF	2.14 A	1.51 A
OFF	OFF	OFF	2.83 A	2.0

### 5.4 Semi-flow function:

Semi-flow function is that there is not step pulse after 200 ms, the driver output current automatically reduced to 40% of rated output current, which is used to prevent motor heat.

## 6. Pins of motor & power:

Connect Makeblock 2H Microstep Driver with Stepper motor 42BYGHW609.

Pins	Motor wiring & Power supply
A+	Black Wire
A-	Green Wire
B+	Red Wire
B-	Blue Wire
DC+	Power supply : DC12-36VDC. The peak input current cannot be up to 2A
DC-	

## 7. Adjustment of troubleshooting

### 7.1 The status on light's indication

PWR: green, normal work light.

ALM: red, failure light, the motor with phase short-circuit, overvoltage and under-voltage protection.

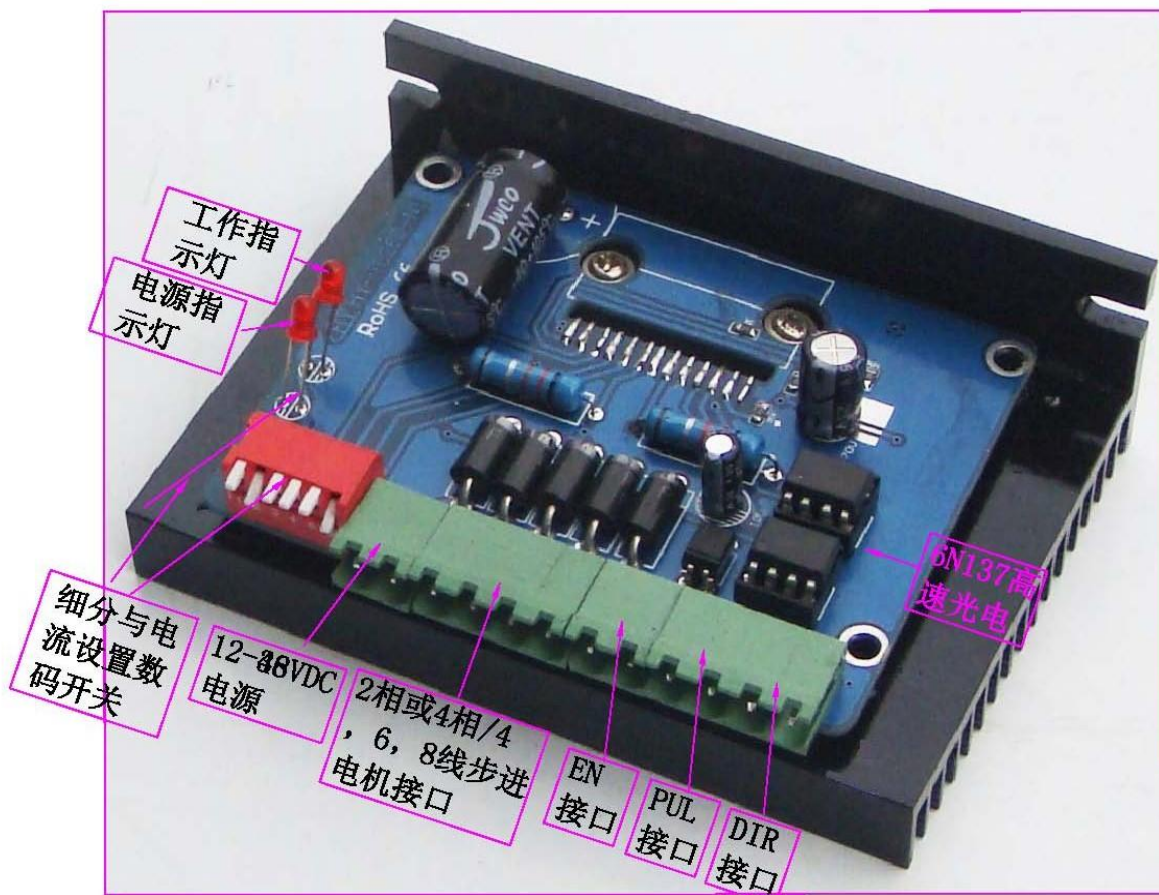
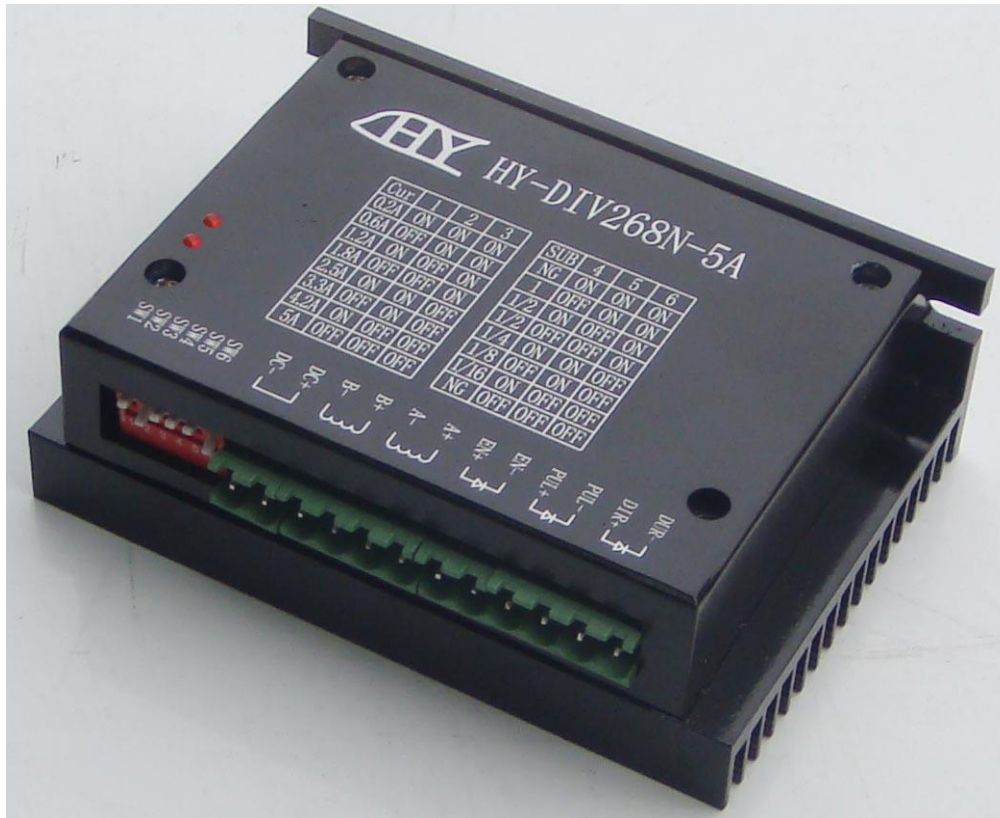
### 7.2 Troubles

Alarm indicator	Reasons	Measures
LED off turn	Wrong connection for power	Check wiring of power
	Low voltages for power	Enlarge voltage of power
Motor doesn't run, without holding torque	Wrong connection of stepper motor	Correct its wiring
	RESET signal is effective when offline	Make RESET ineffective
Motor doesn't run, but maintains holding torque	Without input pulse signal	Adjust PMW & signal level
Motor runs wrong direction	Wrong wires' connection	Change connection for any of 2 wires
	Wrong input direction signal	Change direction setting
Motor's holding torque is too small	Too small relative to current setting	Correct rated current setting
	Acceleration is too fast	Reduce the acceleration
	Motor stalls	Rule out mechanical failure
	Driver does not match with the motor	Change a suitable driver

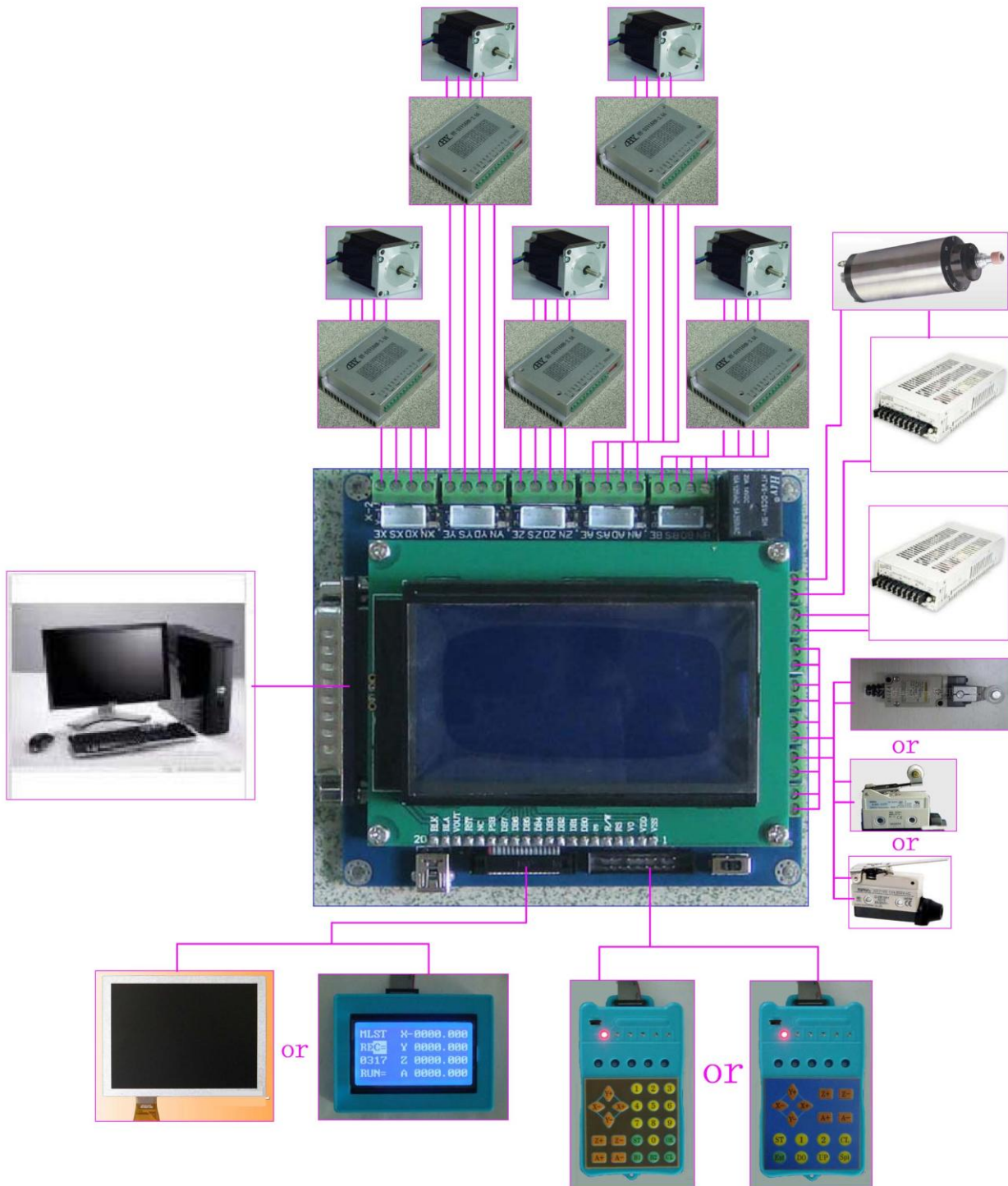
## HY-DIV268N-5A two phase hybrid stepper motor drive manual

Product Link:

<http://www.thanksbuyer.com/cnc-single-axis-tb6600-0-2-5a-two-phase-hybrid-stepper-motor-driver-controller-24891>







—, an overview

HY-DIV268N-5A subdivision-type two-phase hybrid stepping motor drive using DC 12 ~ 48V power supply, suitable for drive

Two-phase hybrid stepping motor dynamic voltage 12 to 48V, the current is less than 5A outer diameter of 35 to 86 mm. This drive using the drive's current loop subdivision control, the motor torque ripple is very small, low-speed running is very smooth, almost no vibration and noise. High-speed torque is much higher than other two-phase drive, high positioning accuracy. Widely used in the engraving machine, CNC machine tools, packaging machinery and other high resolution requirements on the device.

The main features

- 1 average current control, two-phase sinusoidal current drive output
- 2 DC 12 ~ 48V power supply, the internal integration of 12V and 5V regulator
- 3 optically isolated signal input / output
- 4, overvoltage, undervoltage, overcurrent, and white short-circuit protection

5 4 file segmentation and automatic half-streaming capabilities up to 16 segments

4 file output phase current settings

7 high starting speed

8 high-speed torque

—, the electrical parameters

Input voltage DC 12 ~ 48V input

The input current of 1 to 5 amps, select the drive a stepper motor.

Output current of 0.2A ~ 5A

Temperature Operating Temperature -10 to 45 °C; Storage temperature -40 °C to 70 °C

Humidity not condensing, not drops

Gas prohibit combustible gas and conductive dust

Weight 200 grams

## 二、Control signal interface

Figure 1 is a wiring schematic of the drive

1, the definition of control signals

PUL +: step pulse signal is input side or the positive pulse signal input positive terminal

PUL -: the negative input of the negative input pulse signal or a positive pulse signal

DIR +: stepping direction signal input to the positive terminal or negative pulse signal input to the positive terminal

DIR -: stepping direction signal input of the negative side or reverse step pulse signal input negative terminal

EN +: offline can reset signal input side is

EN -: offline can reset signal input negative terminal

Offline enable signal is active, reset drive failure to prohibit any pulse, the output of the drive

Power component is turned off, the motor holding torque.

### 2、Control signal connections

PC control signal can be high, also can be low effective. When active high, the control signal

The negative side together as a signal to active low, positive side of all control signals together as a signal common.

For example, open-collector and PNP output interface circuit diagram is as follows:

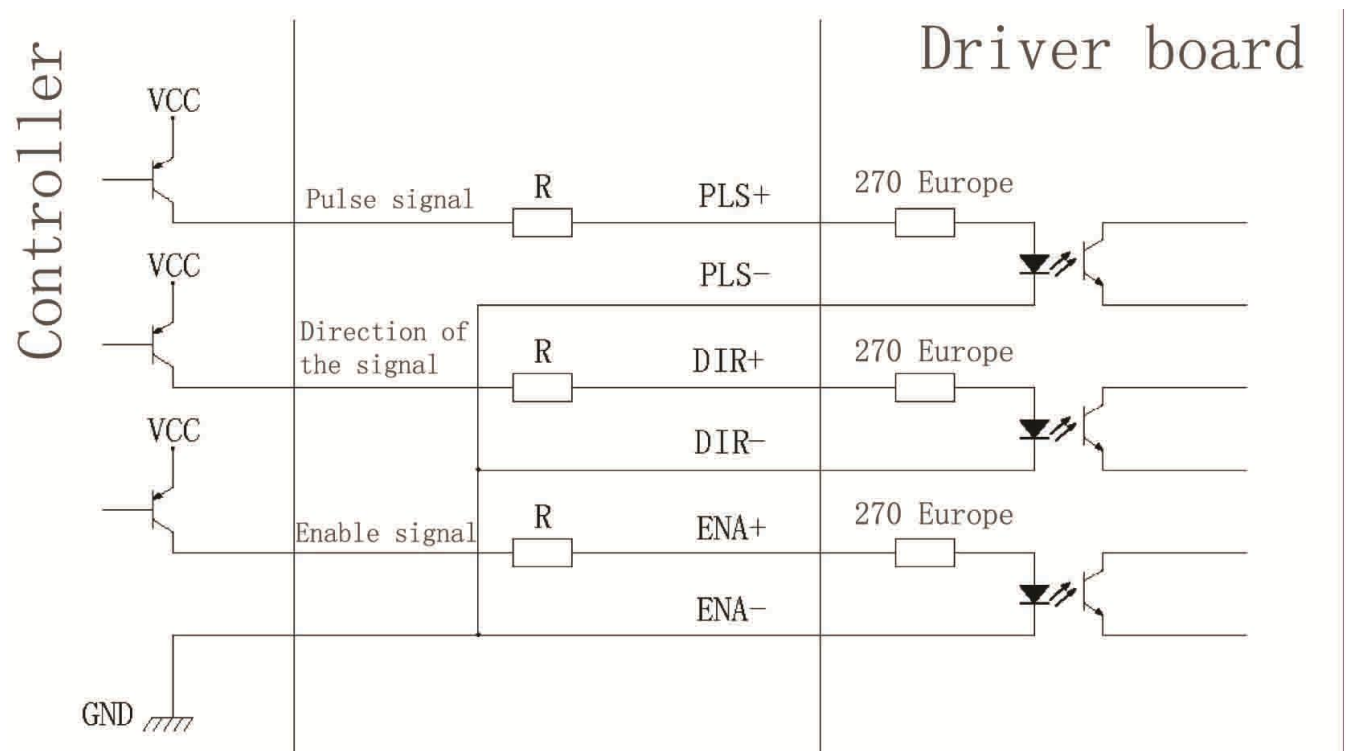


Figure 2. Input interface circuit (common cathode connection)

Note: the VCC is 5V, R short;

VCC value of 12V, R 1K, more than 1/8W resistor;

VCC value of 24V, R 2K, more than 1/8W resistor;

**R Must be connected to the controller output terminals.**

### 三、Function selection (DIP switch on the drive panel)

#### 1、Set the motor per revolution steps

The drive to set the number of steps per motor revolution is 200 (whole step), 400 (2 segments), 1600 (8 segments), 3200 steps (16 segments).

The user can drive the front panel DIP switch SW3 is SW4 bit of the drive to set the number of steps (such as Table 1) :

Table 1

#### 2、Set the output phase current

To drive torque stepper motor, the user can drive panel DIP switch

SW1, SW2 is to set the drive output phase current (RMS) unit amperes, the switch position

Corresponding to the output current, output current value corresponding different types of drive.

Concrete are shown in Table 2.

Output Current (A

Table 2

#### 3、Current setting

细分	1	2	3	电流	4	5	6
NG	ON	ON	ON	0.2A	ON	ON	ON
1	OFF	ON	ON	0.6A	OFF	ON	ON
1/2	ON	OFF	ON	1.2A	ON	OFF	ON
1/2	OFF	OFF	ON	1.8A	OFF	OFF	ON
1/4	ON	ON	OFF	2.5A	ON	ON	OFF
1/8	OFF	ON	OFF	3.3A	OFF	ON	OFF
1/16	ON	OFF	OFF	4.2A	ON	OFF	OFF
NG	OFF	OFF	OFF	5A	OFF	OFF	OFF

#### 4, the semi-flow functionality

The semi-flow function is a step pulse 200ms, the driver output current is automatically reduced to the rated output current

50%, used to prevent motor heating.

Fourth, the power interface

1, DC +, DC-: to connect the drive power

DC +: DC power level, power supply voltage DC 12 to 48V. The maximum current is 5A. The



DC-: DC power supply negative level.

2, A + A-B + B-: to connect the two-phase hybrid stepping motor

Drive and two-phase hybrid stepper motor's connection with the four-wire system, the motor windings in parallel and series connection, and connection method, high-speed performance, but the drive current is large (for motor winding current of 1.73 times), Connected in series when the drive current is equal to the motor winding current.

Five, the installation

Around to have 20mm of space, can not be placed next to the other heating equipment, to avoid dust, oil mist,

Corrosive gases, high humidity is too big and strong vibration. +

Figure 3.

Six fault diagnosis

1, the status light indicates

RUN: green light in the normal work.

ERR: red light, power, light, power indicator

### 11.3 Codis

Els tres primers codis que es mostren a continuació estan escrits en el llenguatge C++. El primer dels codis correspon a la funció "delta\_calcForward" que serveix per a realitzar el càlcul de la cinemàtica directa del robot. El segon correspon a les funcions "delta\_calcInverse" i "delta\_calcAngleYZ" que serveixen per a realitzar el càlcul de la cinemàtica inversa del robot. El tercer codi correspon a un exemple de l'extracció de dades per a realitzar l'anàlisi de l'abast del robot, en aquest cas el del pla  $z=-190$ .

El darrer codi està escrit en el llenguatge d'arduino i és el que s'encarrega del moviment del robot. En la primera part es pot veure la configuració inicial amb la qual es configura que controla cada Pin de l'arduino. La segona part és un bucle fet a partir de molts "for", on a cada "for" el robot traça una línia recta (s'ha posat una part del programa a mode d'exemple del funcionament).

```

int delta_calcForward(float theta1, float theta2, float theta3, float &x0, float &y0, float &z0) {
    float t = (e-f)*tan30/2;
    float dtr = pi/(float)180.0;

    theta1 *= dtr;
    theta2 *= dtr;
    theta3 *= dtr;

    float y1 = -(t + rf*cos(theta1));
    float z1 = -rf*sin(theta1);

    float y2 = (t + rf*cos(theta2))*sin30;
    float x2 = y2*tan60;
    float z2 = -rf*sin(theta2);

    float y3 = (t + rf*cos(theta3))*sin30;
    float x3 = -y3*tan60;
    float z3 = -rf*sin(theta3);

    float dnm = (y2-y1)*x3-(y3-y1)*x2;

    float w1 = y1*y1 + z1*z1;
    float w2 = x2*x2 + y2*y2 + z2*z2;
    float w3 = x3*x3 + y3*y3 + z3*z3;

    // x = (a1*z + b1)/dnm
    float a1 = (z2-z1)*(y3-y1)-(z3-z1)*(y2-y1);
    float b1 = -((w2-w1)*(y3-y1)-(w3-w1)*(y2-y1))/2.0;

    // y = (a2*z + b2)/dnm;
    float a2 = -(z2-z1)*x3+(z3-z1)*x2;
    float b2 = ((w2-w1)*x3 - (w3-w1)*x2)/2.0;

    // a*z^2 + b*z + c = 0
    float a = a1*a1 + a2*a2 + dnm*dnm;
    float b = 2*(a1*b1 + a2*(b2-y1*dnm) - z1*dnm*dnm);
    float c = (b2-y1*dnm)*(b2-y1*dnm) + b1*b1 + dnm*dnm*(z1*z1 - re*re);

    // discriminant
    float d = b*b - (float)4.0*a*c;
    if (d < 0) return -1; // non-existing point

    z0 = -(float)0.5*(b+sqrt(d))/a;
    x0 = (a1*z0 + b1)/dnm;
    y0 = (a2*z0 + b2)/dnm;
    return 0;
}

```

```

int delta_calcAngleYZ(float x0, float y0, float z0, float &theta) {
    float y1 = bf;
    y0 -= bm;
    // z = a + b*y
    float a = (x0*x0 + y0*y0 + z0*z0 + rf*rf - re*re - y1*y1)/(2*z0);
    float b = (y1-y0)/z0;
    float d = -(a+b*y1)*(a+b*y1)+rf*(b*b*rf+rf);
    if (d < 0) return -1;
    float yj = (y1 - a*b - sqrt(d))/(b*b + 1);
    float zj = a + b*yj;
    theta = 180.0*atan(-zj/(y1 - yj))/pi + ((yj>y1)?180.0:0.0);
    return 0;
}

int delta_calcInverse(float x0, float y0, float z0, float &theta1, float &theta2, float &theta3) {
    theta1 = theta2 = theta3 = 0;
    int status = delta_calcAngleYZ(x0, y0, z0, theta1);
    if (status == 0) status = delta_calcAngleYZ(x0*cos120 + y0*sin120, y0*cos120-x0*sin120, z0, theta2);
    if (status == 0) status = delta_calcAngleYZ(x0*cos120 - y0*sin120, y0*cos120+x0*sin120, z0, theta3);
    return status;
}

```

```

int main () {
    FILE *pf;
    float theta1=0.0;
    float theta2=0.0;
    float theta3=0.0;
    float y0=0.0;
    float z0=0.0;
    float x0=0.0;
    vector<float> a (3,0);
    vector<vector<float> > > v;
    for (int i=0; i<90; i=i+1){
        for (int j=0; j<90; j=j+1){
            for (int k=0; k<90; k=k+1){
                delta_calcForward(i,j,k,x0,y0,z0);
                if(delta_calcForward(i,j,k,x0,y0,z0)!=-1){
                    a[0]=x0;
                    a[1]=y0;
                    a[2]=z0;
                    if (int(a[2])==-190) {
                        v.push_back(a);
                    }
                }
            }
        }
    }

    pf=fopen("pla190.xls","w");
    int val=0;
    int ymax=0;
    int ymin=0;
    int xmax=0;
    int xmin=0;
    int posxmax=0;
    int posxmin=0;
    int posymax=0;
    int posymin=0;
    for (int i=0; i<v.size(); i=i+1){
        for (int j=0; j<3; j=j+1){
            if (j==0) {
                if ((int)v[i][j]>xmax){
                    posxmax=i;
                    xmax=(int)v[i][j];
                }
                if ((int)v[i][j]<xmin){
                    posxmin=i;
                    xmin=(int)v[i][j];
                }
            }
            if (j==1) {
                if ((int)v[i][j]>ymax){
                    posymax=i;
                    ymax=(int)v[i][j];
                }
                if ((int)v[i][j]<ymin){
                    posymin=i;
                    ymin=(int)v[i][j];
                }
            }
            val=(int)v[i][j];
            fprintf(pf,"%ld",val);
            fprintf(pf,"\t");
        }
        fprintf(pf,"\n");
    }
    fclose(pf);
    cout<<"X max "<<xmax<<" pos max"<<posxmax<<endl;
    cout<<"X min "<<xmin<<" pos min"<<posxmin<<endl;
    cout<<"Y max "<<ymax<<" pos max"<<posymax<<endl;
    cout<<"Y min "<<ymin<<" pos min"<<posymin<<endl;
}

```

```
const int dirPin = 10;
const int stepPin = 9;
const int dirPin2 = 4;
const int stepPin2 = 3;
const int dirPin3 = 6;
const int stepPin3 = 7;
int stepDelay;
//okey
void setup() {
  // Declarar los pines como salida
  delay(10000);
  pinMode(dirPin, OUTPUT);
  pinMode(stepPin, OUTPUT);
  pinMode(dirPin2, OUTPUT);
  pinMode(stepPin2, OUTPUT);
  pinMode(dirPin3, OUTPUT);
  pinMode(stepPin3, OUTPUT);
}
```

```
void loop() {  
  digitalWrite(dirPin, HIGH);  
  digitalWrite(dirPin2, HIGH);  
  digitalWrite(dirPin3, HIGH);  
  stepDelay = 3;  
  for (int x = 0; x < 56; x++) {  
    digitalWrite(stepPin, HIGH);  
    digitalWrite(stepPin2, HIGH);  
    digitalWrite(stepPin3, HIGH);  
    digitalWrite(stepPin, LOW);  
    digitalWrite(stepPin2, LOW);  
    digitalWrite(stepPin3, LOW);  
    delay(20);  
  }  
  digitalWrite(dirPin3, LOW);  
  stepDelay = 3;  
  digitalWrite(dirPin3, LOW);  
  stepDelay = 3;  
  for (int x = 0; x < 72; x++) {  
    if (x<16){  
      digitalWrite(stepPin, HIGH);  
    }  
    digitalWrite(stepPin2, HIGH);  
    if (x<48){  
      digitalWrite(stepPin3, HIGH);  
    }  
    if (x<16) {  
      digitalWrite(stepPin, LOW);  
    }  
    digitalWrite(stepPin2, LOW);  
    if (x<48){  
      digitalWrite(stepPin3, LOW);  
    }  
    delay(20);  
  }  
  digitalWrite(dirPin, LOW);  
  digitalWrite(dirPin2, LOW);  
  stepDelay = 3;  
}
```